Pulse Instruments

FOCAL PLANE TEST LAB

PUL

OPERATING MANUAL

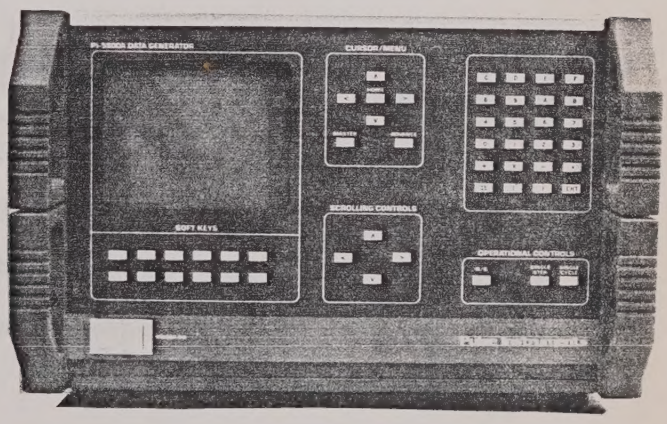
Model PI-5800A Data Generator

58101 Non-Volatile Memory Option

58102 Extended Level Looping Option

58002 Programmable Delay Option

Manual #1



PULSE INSTRUMENTS COMPANY
1234 Francisco Street
Torrance, California 90502
(213) 515-5330

Part Number 91000160

Revision D

Dated July 15, 1986

1.0 WARRANTY AND SAFETY INFORMATION

1.1 WARRANTY

Pulse Instruments warrants these products to be free from defects in material and workmanship for a period of one year from date of shipment.

During the warranty period, Pulse Instruments will, at its own discretion, repair or replace any products or materials returned, freight prepaid, which prove to be defective through normal use.

This warranty does not apply to defects resulting from unauthorized modification, misuse, or neglect. This warranty is in lieu of all other warranties expressed or implied, and Pulse Instruments shall not be liable for any consequential damages resulting from the use of this product.

1.2 SAFETY INFORMATION

1.2.1 General

1. The PI-5800A Data Generator uses a detachable 3-wire power cord for connection to both the power source and to earth ground. This unit is to be connected only to a power source outlet that has provision for properly grounding the plug.

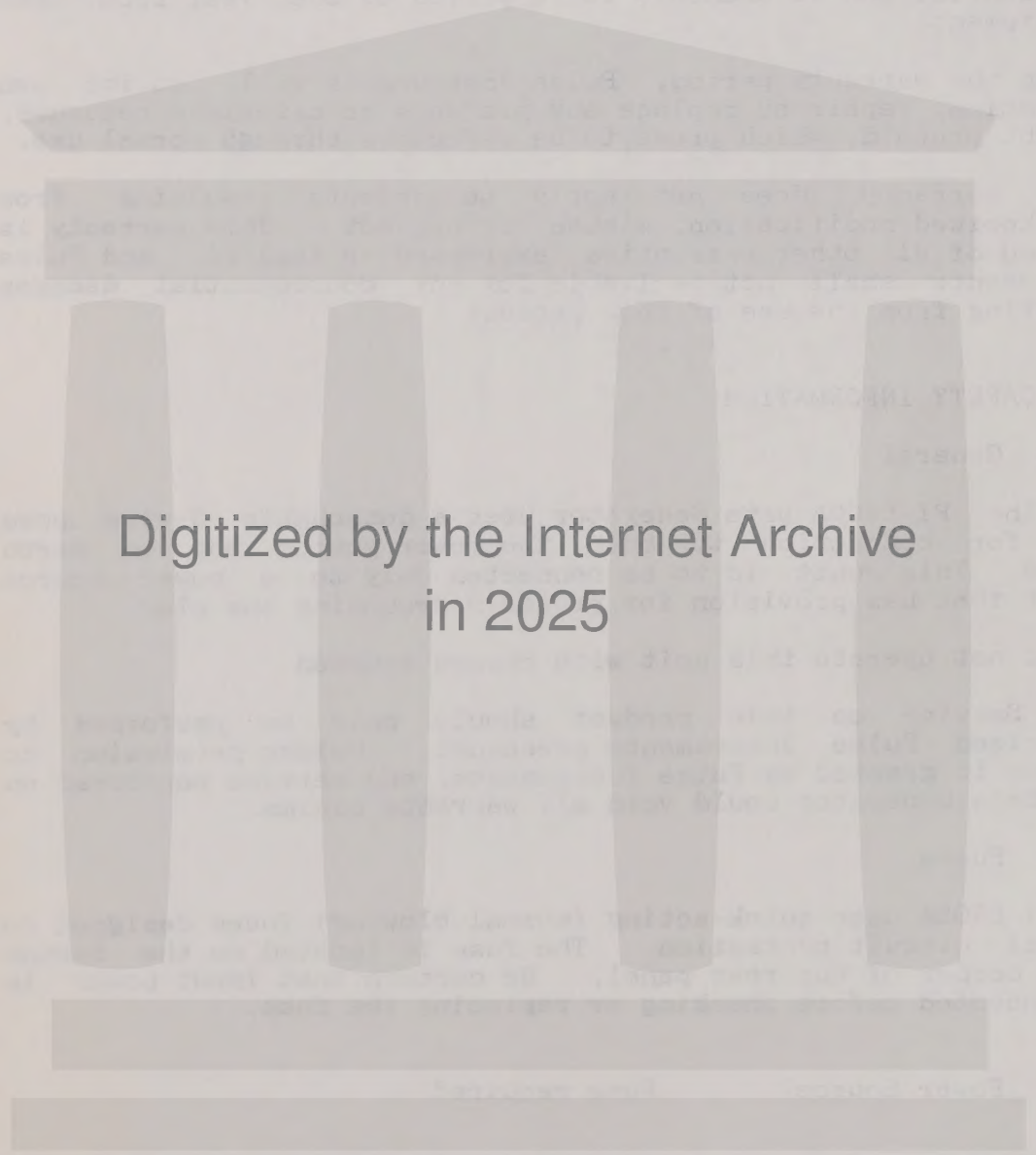
2. Do not operate this unit with covers removed.

3. Service on this product should only be performed by authorized Pulse Instruments personnel. Unless permission to service is granted by Pulse Instruments, any service performed on this Data Generator could void all warranty claims.

1.2.2 Fuses

The PI-5800A uses quick-acting (normal blowing) fuses designed for general circuit protection. The fuse is located on the bottom right corner of the rear panel. Be certain that input power is disconnected before checking or replacing the fuse.

Power Source	Fuse required
120V.	3AG "Normal-Blo" 8 amp 125V. Littlefuse 312008 or equivalent.
230V.	3AG "Normal-Blo" 4 amp 250V. Littlefuse 312004 or equivalent.


Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_11224

2.0 TABLE OF CONTENTS

2.1 TABLE OF CONTENTS

1. WARRANTY AND SAFETY INFORMATION 2

1.1 WARRANTY 2

1.2 SAFETY INFORMATION 2

1.2.1 General 2

1.2.2 Fuses 2

2.0 TABLE OF CONTENTS 3

2.1 TABLE OF CONTENTS 3

2.2 LIST OF ILLUSTRATIONS 9

3.0 SPECIFICATIONS 10

3.1 OUTPUT CHARACTERISTICS 10

3.2 INPUT CHARACTERISTICS 10

3.3 TRIGGER OUTPUT CHARACTERISTICS 10

3.4 GENERAL 10

4.0 INSTALLATION 12

4.1 INTRODUCTION 12

4.2 PACKING AND UNPACKING 12

4.3 POWER REQUIREMENTS 12

4.4 COOLING AND VENTILATION 13

4.5 CONFIGURATION 13

4.6 REAR PANEL CONNECTIONS 14

4.7 CONNECTION OF OUTPUT CABLES 16

5.0 BASIC DESCRIPTION 17

6.0 ORDERING INFORMATION AND OPTIONS 18

7.0 FRONT PANEL CONTROLS 19
7.1 GENERAL.....19
7.1.1 Power On/Off Switch 19
7.1.2 Cursor Controls 19
7.1.3 Home Key 19
7.1.4 Master Key 19
7.1.5 Advance Key 19
7.1.6 Scrolling Controls 19
7.2 OPERATIONAL CONTROLS 19
7.2.1 R/S Key 19
7.2.2 Reset Key 19
7.2.3 Single Cycle Key 20
7.3 SOFTKEYS 20
7.3.1 General 20
7.4 DATA KEYS 21
7.4.1 Data Keys 21
7.4.2 Clear Entry 21
7.4.3 Enter 21
8.0 SYSTEM MENUS 22
8.1 CONFIGURATION DISPLAY 22
8.1.1 General 22
8.2 MASTER MENU 23
8.2.1 General 23
8.2.2 Explanation of Softkeys 24

8.3 PATTERN ENTRY AND EDITING MENU 25
8.3.1 General 25
8.3.2 Subpatterns 26
8.3.3 Setting Up Subpatterns 27
8.3.4 Loading Data into Subpatterns 28
8.3.5 Explanation of Softkeys.....29
8.3.5.1 Page 1 Softkeys.....29
8.3.5.2 Page 2 Softkeys.....31
8.3.6 Editing.....32
8.3.7 Algorithmic Data Entry.....32
8.3.8 Cascade Mode..... 33
8.3.9 Dynamic Update (synchronous).....33
8.3.10 Update (asynchronous).....34
8.4 INSTRUCTION ENTRY AND EDITING MENU 35
8.4.1 General 35
8.4.2 Explanation of Softkeys 36
8.4.3 Page 2 Softkeys.....37
8.4.4 Instruction Programming.....38
8.5 OUTPUT FORMAT AND CHANNEL STATUS MENU 40
8.5.1 General 40
8.5.2 Explanation of Softkeys 41
8.5.3 Output Formats 41
8.5.4 Mixed-Mode Channel Formatting 42
8.5.5 Pulse Width in the RZ Mode.....42

8.6 CLOCK AND INTERFACE SET-UP MENU 43
8.6.1 General 43
8.6.2 Explanation of Softkeys 44
8.6.3 Setting the Internal Clock Period 44
8.6.4 External Clock Input 45
8.7 WAVEFORM DISPLAY MENU 46
8.7.1 General 46
8.7.2 Explanation of Softkeys 46
8.8 SELF TEST MENU 47
8.8.1 General Description 47
8.8.2 Explanation of Softkeys 47
8.9 HIGH RESOLUTION CHANNEL DELAY MENU 48
8.9.1 General Description 48
8.9.2 Setting the Programmable Delay Over the IEEE-488 Bus 49
8.9.3 Explanation of Softkeys 49
9.0 QUICK SET-UP AND USE 50
10.0 OPTIONS FOR THE PI-5800A 53
10.1 58102 Extended Level Looping Option 53
10.1.1 General 53
10.1.2 Third Level Looping 53
10.1.3 Conditional Branching 56
10.2 58101 Non-Volatile Memory Option 59
10.2.1 General 59

10.2 58101 Non-Volatile Memory Option (Cont'd)

10.2.2 Explanation of Softkeys - 58102 Non-Volatile Memory Option 60

10.2.3 Clearing the Entire 58102 Memory Option - Internal Memory 61

10.2.4 Store Function 61

10.2.5 Loading Programs from the Non-Volatile Memory Option 62

10.2.6 To Clear Individual Programs from the Non-Volatile Memory 63

10.2.7 To Change Begin Numbers 63

10.2.8 Non-Volatile Memory Conservation 64

11.0 PROGRAMMING THE PI-5800A 66

11.1 GENERAL 66

11.2 BINARY DATA ENTRY 67

11.3 PROGRAMMING COMMANDS 68

11.4 EDITING COMMANDS 73

11.4.1 Programming Mixed Format Modes 73

11.5 SERVICE REQUEST (SRQ) 75

11.6 SAMPLE PROGRAM FOR PI-5800A (written in BASIC) ...77

11.7 PLACING THE DATA GENERATOR IN THE DYNAMIC UPDATE CONDITION 79

11.8 SETTING THE PROGRAMMABLE DELAY OVER THE IEEE-488 BUS 80

11.9 PROGRAMMING THE 58102 EXTENDED LEVEL LOOPING OPTION 80

12.0 ERROR AND WARNING MESSAGES 81
12.1 GENERAL 81
12.2 EXPLANATION OF ERROR AND WARNING MESSAGES 81
13.0 RS-232-C COMPATIBILITY 85
13.1 GENERAL 85
13.2 RS-232-C PIN OUTS 85
13.3 PROGRAMMING INFORMATION 85
14.0 REPLACING EPROMS IN THE PI-5800A 87
14.1 GENERAL 87

2.2 LIST OF ILLUSTRATIONS

FIGURE NUMBER

1. PI-5800A Rear View 14
2. Output Connectors 16
3. Configuration Display 22
4. Master Menu 23
5. Pattern Entry and Editing Menu 25
6. Defined Subpattern 26
7. Instruction Entry and Editing Menu 35
8. Output Format and Channel Status Menu 40
9. Clock and Interface Set-Up Menu 43
10. Output Waveform Display Menu 46
11. Self Test Menu 47
12. High Resolution Channel Delay Menu 48
13. The 8 Bits of the Service Request Byte 75

3.0 SPECIFICATIONS

PI-5800A DATA GENERATOR SPECIFICATIONS

3.1 OUTPUT CHARACTERISTICS[1]

Output Resistance	Back terminated 50 ohms ± 10%
Pulse Duration	20ns minimum NRZ mode, 10ns RZ mode
Fan Out:	4 TTL loads minimum for duration ≥20ns
	2 TTL loads minimum for duration <20ns
Skew:	3ns maximum between any 2 channels on the same 16-channel card
	4ns maximum between any 2 channel within the same 64-channel unit
Rise and Fall Time:	tr ≤8ns, tf ≤6ns
High Level:	3.5V. minimum, 15ns after rising edge has reached 1.5V
	3V minimum for duration <20ns
Low Level:	0.45V maximum for duration ≥20ns
	0.6V maximum for duration <20ns
High Impedance	Disable Time2: tphz ≤16ns, tplh ≤16ns
	Enable Time2: tpzh ≤23ns, tpzl ≤23ns
Channels	64 per mainframe maximum in 16 channel increments

3.2 INPUT CHARACTERISTICS

External Clock:	TTL compatible, 2.4V, 10ns pulse width minimum required
Cascade Clock:	TTL compatible, to be driven by cascading clock output only
External Halt:	TTL compatible, 2.4V, 6ns pulse width minimum required. Positive going or negative going slope selectable by rear panel mounted switch

3.3 TRIGGER OUTPUT CHARACTERISTICS[1]

Trigger:	TTL compatible. Fan Out = 2TTL loads
Programmable Trigger	TTL compatible. Fan Out = 4TTL loads for duration ≥20ns
	Fan Out = 2TTL loads for duration ≤20ns

3.4 GENERAL

Internal Clock Freq.:	1KHz to 50MHz in 20ns increments
Instruction Mem. Depth:	256 lines per program (Begin Statement) up to a maximum of 999 lines, all programs combined
Display:	7-inch CRT, 80 characters per line

Memory:	4,096 bits per channel
CPU:	Z80A
Prog. Delay (Option)	500ps to 31.5ns in 500ps increments ±300ps
Compatibililty:	IEEE-488 GPIB, RS-232C
Power PI-5800A	110 vac ±10%, 8 amps max. 47/63 Hz 220 vac ±10%, 4 amps max. 47/63 Hz
Dimensions:	10.5" H x 17.5" W x 23.75" D
Weight:	49-55 lbs. net
Operating Temperature:	0 deg. C to 50 deg. C
Warranty:	One year parts and labor from date of shipment (See Page 1)

Note:

1. Unless otherwise specified, all dynamic measurements are made using a 3 ft. length, 50 ohm coaxial cable transmitted into 15pf/1M ohm (such as the input of a 300MHz oscilloscope).

2. R1=R2=500 ohm, CL=90pf, including 3 ft. of cable capacitance and scope. (See TI TTL Data Book, Volume 3).

4.0 INSTALLATION

4.1 INTRODUCTION

This operating manual contains all the information required to install, operate and program the PI-5800A Data Generator, and all options and accessories. Unless otherwise noted, all references to the PI-5800 are the same for the PI-5800A.

The PI-5800A Data Generator is a complex instrument that is capable of performing many tasks. To use this instrument to full advantage, a thorough reading of this manual is required. However, IF YOU WOULD PREFER AT PRESENT TO QUICKLY GAIN ONLY THE BASIC OPERATING KNOWLEDGE TO USE THE INSTRUMENT, PLEASE REFER TO THE SECTION ON QUICK SET UP, Section 9.0.

4.2 PACKING AND UNPACKING

The PI-5800A Data Generator was shipped to you in a foam filled carton that was designed to withstand normal shipping requirements. Please report any damages to the carrier as soon as possible.

The PI-5800A Data Generator was shipped with the following accessories:

1 - Operator's Manual
1 - Mating connector for each 16-channel PC card (the cards are installed in unit)
1 - Power Cord

Please contact Pulse Instruments immediately if you find any discrepancies.

***** DO NOT TURN THE POWER ON UNTIL *****
***** YOU HAVE READ THE FOLLOWING: *****

4.3 POWER REQUIREMENTS

The Pulse Instruments Model PI-5800A Data Generator is designed to operate from either a 120V or 220V power source, depending upon the model shipped from the factory. The label in the bottom rear corner of the back panel above the power receptacle indicates the voltage input required for operation of your unit. Please verify that the voltage indicated is correct for your intended application before connecting the power cord to the unit or to the power source.

4.4 COOLING AND VENTILATION

To prevent damage or over-heating of your Data Generator, please make certain that there are no obstructions near the unit that would tend to defeat the operation of the rear fans or the air flow vents located around the unit. Adequate circulation and ventilation are necessary for proper unit operation and instrument life.

4.5 CONFIGURATION

The PI-5800A is normally supplied in rackmount configuration. If you intend to use the Data Generator for bench top applications only, the factory should be notified at the time of ordering, as removal of the rackmount slides require operations from inside the unit.

All printed circuit boards that are supplied as part of your unit were installed at the factory. This is verified by the configuration display which is visible on the CRT immediately after power up. To remove this display and get the Master Menu, press any key.

4.6 REAR PANEL CONNECTIONS

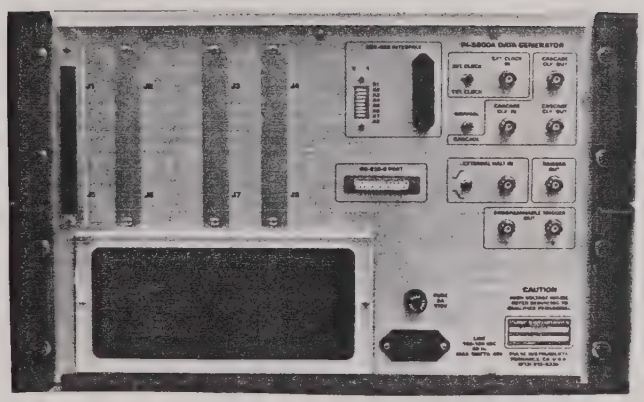


Figure 1. PI-5800A Rear View

INT Clock.....When clock switch is placed in this position, the internal clock within the PI-5800A is selected.

EXT Clock .....Allows use of any TTL clock input from D.C. to 50MHz with 10 nanoseconds minimum pulse width.

Cascade Clock.....Used as an input from one of the cascade clock outputs to synchronize PI-5800A to another mainframe. This input should not be used as an external clock source, but should be used in conjunction with cascade clock only.

Normal.....Switch setting when either internal or external clock mode is selected as opposed to Cascade Clock which is selected when synchronization of 2 PI-5800A Data Generators is required.

External Halt.....Immediately halts internal clock, can be set to halt on positive or negative level by setting its associated switch.

Cascade Clocks.....Used for supplying synchronized clock outputs.

Prog. Trig. 1.....Allows use of Channel A-1 as a trigger output.

Prog. Trig. 2.....Allows use of Channel A-2 as a trigger output.

Trigger Out.....Supplies TTL trigger pulse upon occurance of each programmed begin statement.

RS-232C Port.....EIA standard serial interface.

IEEE-488 Interface.....Interface to general purpose instrument bus. IEEE-488 Std. 1978

Power Connector.....Accepts supplied 3 wire cord for connection to power source.

J1 - J4.....16-Channel output connectors. See Section 4.7.

J5 - J8.....These connectors are to be used in future configurations and have no purpose at this time. See Section 4.7.

58102 EXTENDED LEVEL LOOPING OPTION FLAGS

Ground*.....Pin 18 on RS-232C connector
A Flag.....Pin 19 on RS-232C connector
B Flag.....Pin 24 on RS-232C connector
Ground*.....Pin 25 on RS-232C connector

*These grounds are tied together internally.

RS-232-C Port

Pin 1 - Protective Ground (shield)

Pin 2 - Receive Data (xmit data connection on controller)

Pin 3 - Transmit Data (receive data connection on controller)

Pin 5 - Clear to Send

Pin 6 - Data Set Ready

Pin 7 - Signal Ground

4.7 CONNECTION OF OUTPUT CABLES

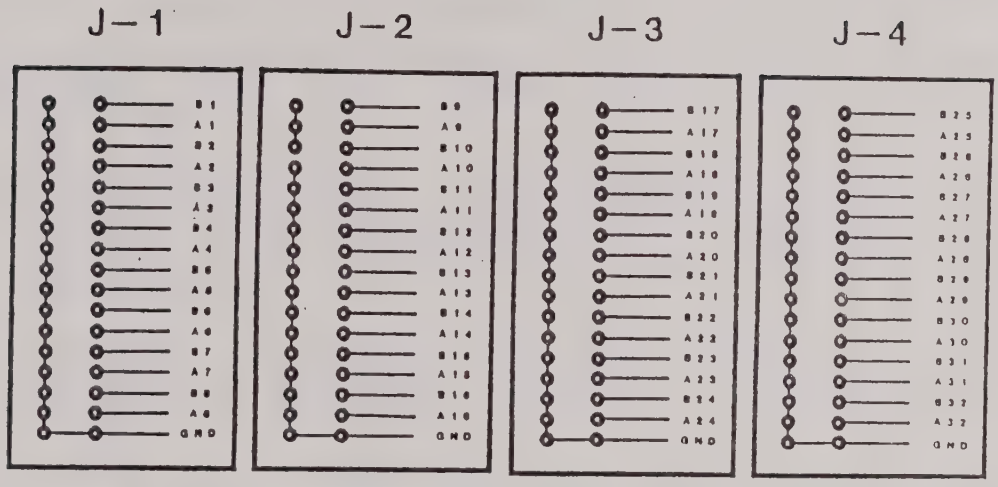


Figure 2. Output Connectors

The 16-channel output cables are connected to J1 through J4 as follows:

J1 Channels A1 through B8
J2 Channels A9 through B16
J3 Channels A17 through B24
J4 Channels A25 through B32

You were supplied with a mating connector for each 16-channel Pattern card supplied. The above drawing provides the output channel pin-outs. It is recommended that all outputs be wired with 50 ohm coaxial cable equivalent to RG 178B/U. Also, cables should not exceed 4 feet in length or serious signal degradation could occur at high frequencies.

If you were supplied with Pulse Instruments cables 88000050 or 88000040, you can connect the cables to the J1 through J4 connectors. The connectors are keyed, and can only be inserted one way. J5 through J8 connectors have no purpose at this time.

5.0 BASIC DESCRIPTION

With true DC to 50 MHz operation, in all modes, and a memory depth of 4K bits for each of 16 to 64 channels, the PI-5800A Data Generator is a versatile digital signal source that can generate extremely long patterns. The instrument is also easy to program.

Most of the programming required for operation of the PI-5800A is accomplished with only 2 menus which cover Pattern and Instruction Entry.

Enhancing the instrument's versatility are the selective output formats, which allow the choice of RZ (return to zero), or NRZ (not return to zero), high empedance or 50 ohm output, and true or complement outputs, or mixtures of these formats on same or different channels.

Since patterns can be updated dynamically during run, as much as 99% of a channel's data output can be synchronously changed while the PI-5800A is in the Run mode.

Each channel has a memory depth of 4K bits with 2 levels of looping. As many as 256 repeatable subpatterns may be repeated within the main program to produce patterns exceeding 250 billion bits in length.

The PI-5800A contains two separate sets of random access memory which work in parallel to provide you with pattern generation, while at the same time allowing you to perform other functions through use of the keyboard and CRT, without disturbing the pattern output.

Should you desire to change the pattern output, it can be accomplished easily during the run mode by simply making the pattern changes on the Pattern Entry and Editing Menu, and then pressing the [UPDATE] softkey. The output pattern then contains the changes that were made without stopping the unit to reprogram.

One additional PI-5800A mainframe containing four 16-channel data output cards can expand the system to a maximum of 128 channels. Other features of the instrument include output levels of 0 to 3.5V, back-terminated by 50 ohms; an external clock input of DC to 50 MHz; IEEE-488 or RS-232C compatibility; and optional EEPROM program storage. In addition, the instrument accepts optional 16-channel cards, whose delay is programmable in 500 psec increments.

6.0 CONFIGURATION INFORMATION AND OPTIONS

PI-5800A Data Generator.....Mainframe requires printed circuit cards listed below to complete system. Mainframe can accommodate 4 printed circuit cards. For systems requiring more than 64 channels, another PI-5800A mainframe is required for the additional channels. The PI-5800A is supplied in rackmount configuration, unless otherwise specified.

58001.....16-Channel output printed circuit card, 4K bits memory per channel.

58002.....16-Channel output printed circuit card, with programmable delay from 500 picoseconds to 31.5 nanoseconds in 500 picosecond increments.

88000040.....IEEE-488 standard cable, 6 feet in length.

88000050.....16-Channel output, 50 ohm coaxial ribbon cable, 4 feet in length, BNC terminated.

88000060.....16-Channel output, 50 ohm coaxial ribbon cable, 4 feet in length, mass terminated both ends.

58101 Non-Volatile Memory Option.....Allows user to store and recall up to 48K bytes of pattern data and 1000 lines of instrument set-up and pattern data.

58102 Extended Level Looping Option.....Allows further repeat of existing repetition factors to greatly extend pattern looping and nesting capabilities. The option also includes conditional branches that are accessed by hardware and software flags.

7.0 FRONT PANEL CONTROLS

7.1 GENERAL

***** NOTE: WORDS IN THIS MANUAL WHICH ARE BOTH BRACKETED *****
***** AND UNDERSCORED REPRESENT FRONT PANEL KEYS, EG. THE *****
***** THE MASTER KEY IS REPRESENTED BY [MASTER]. *****

7.1.1 Power On/Off Switch.....Applies power to the Data Generator; turning power off will remove all contents of the Volatile Memory.

7.1.2 Cursor Controls.....Allows movement of the cursor up, down, left or right. (Cursor movement on each menu is based on individual menu requirements.)

7.1.3 [HOME] Key.....On the Pattern Entry and Editing Menu, pressing the home key once, positions the cursor at bit position one of the cursor's present channel location. The next depression will locate the cursor at bit position one of channel "A1".

7.1.4 [MASTER] Key.....Calls up the directory of all system menus on the CRT.

7.1.5 [ADVANCE] Key.....Increments through the menu sequence to allow selection of the required menu.

7.1.6 Scrolling Controls.....Allows scrolling the CRT screen format up, down, left or right, based on individual menu requirement.

7.2 OPERATIONAL CONTROLS

7.2.1 [R/S] Key.....The [RUN/STOP] key is used to run any program that you want to automatically keep repeating. The unit will continuously run the previously compiled program when the [R/S] key places the unit in the Run mode. The [SINGLE CYCLE] key will run the compiled program one time. The number following the blinking run statement on the CRT during the Run mode references the program that is currently compiled and running.

7.2.2 [SINGLE STEP] Key.....The Single Step mode allows you to step through a pre-selected number of clocks. In order to use this mode, any normal program must be entered and compiled (see Section 8.4.2, [COMPILE] softkey), and the Data Generator must be in the Stop mode. To run the Single Step mode, stop the program, recompile, and press the [SINGLE STEP] key in the operational controls group of keys. This can be done on any menu display.

The softkeys will now display as follows:

[1CLOCK] [10 CLOCKS] [100 CLOCKS] [1000 CLOCKS] [EXIT]

By pressing the desired softkeys you can now cycle through any desired number of clocks. Pressing the [EXIT] key will bring back the original softkey displays and exit the Data Generator from the Single Step mode.

7.2.3 [SINGLE CYCLE] Key.....In the Single Cycle mode, the PREVIOUSLY COMPILED PROGRAM is run, one time through, and the Data Generator stops. You can then re-run the program by pressing the [SINGLE CYCLE] key again. The [R/S] key allows you to run the pattern continuously, but the unit must be re-compiled before implementing the [R/S] function.

7.3 SOFTKEYS

7.3.1 General

The function and value of the 12 softkeys are software defined. Since each successive menu redefines their use, the softkeys represent approximately 84 different functions. Their location directly below the CRT allows easy coordination with their current value or function, which is displayed on the lower portion of the CRT. Please refer to the menu section for the value and function descriptions of the individual softkeys.

7.4 DATA KEYS

7.4.1 Data Keys

The data keys enable new parameter values and output data to be entered. The PI5800A will accept binary entry.

Binary data is entered with the [1] and [0] keys. Please refer also to Algorithmic Data Entry, Section 8.3.7.

7.4.2 Clear Entry [CE]

The [CE] key will clear out any information that has been entered by the data keys prior to pressing [ENT].

7.4.3 Enter [ENT]

The [ENT] key assembles all new parameter values and performs the entry of these values into program operation.

8.0 SYSTEM MENUS

8.1 CONFIGURATION DISPLAY

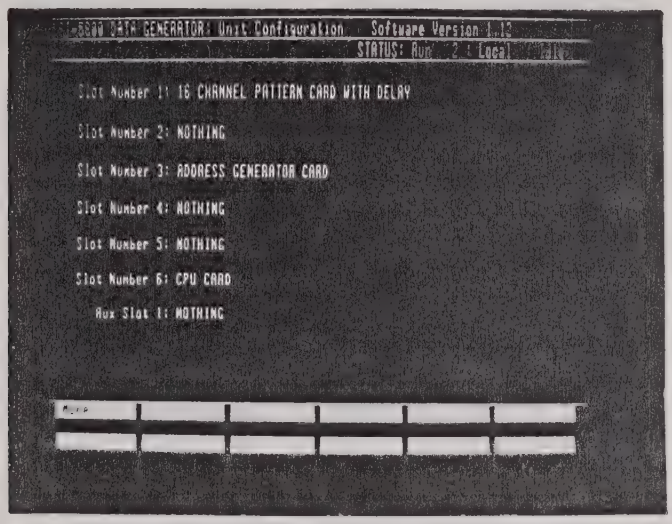

5800 Data GENERATOR: Unit Configuration Software Version 1.13
STATUS: Run 2 1 Local
Slot Number 1: 16 CHANNEL PATTERN CARD WITH DELAY
Slot Number 2: NOTHING
Slot Number 3: ADDRESS GENERATOR CARD
Slot Number 4: NOTHING
Slot Number 5: NOTHING
Slot Number 6: CPU CARD
Aux Slot 1: NOTHING

Figure 3. Configuration Display

8.1.1 General

Immediately upon Power-up, the PI-5800A Data Generator CRT will show the configuration display. This display shows the six internal card slots and lists their contents. Refering to this menu will let you know how many channels of pattern output are available, which channel cards have programmable delay, or if there are other optional features in the particular unit. Press any key to clear this display and call up the Master Menu.

The Configuratiion Display can be called at any time by pressing the [MASTER] key to display the Master Menu. To view the Configuration Display, press the [UNIT CONFIG] softkey, on the Master Menu.

NOTE: When the Configuration Display is called at any time except at Power-up, the unit must be compiled and updated before running. See Sections 8.3.10 and 8.4.2 for further information on these commands.

8.2 MASTER MENU

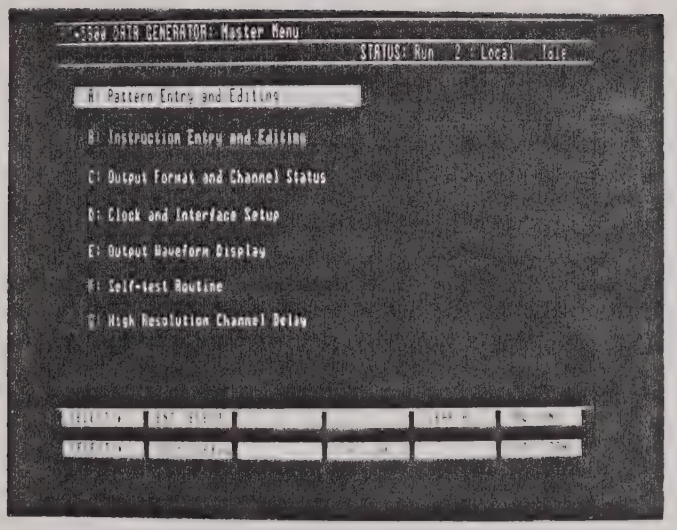

STATUS: Run
A: Pattern Entry and Editing
B: Instruction Entry and Editing
C: Output Format and Channel Status
D: Clock and Interface Setup
E: Output Waveform Display
F: Self-test Routine
G: High Resolution Channel Delay

Figure 4. Master Menu

8.2.1 General

There are eight menus that can be called in the PI-5800A Data Generator. The Master Menu is the directory of all system menus. Once an operating menu has been selected from the Master Menu display, the [SELECT UP] or [SELECT DOWN] softkeys are used to place the cursor on the desired menu selection. Pressing the [ENT SELECT] softkey will display your selection. As an alternative, the [ADVANCE] key can be pressed to cycle through the menus to get the required display.

The system menus are:

-Pattern Entry and Editing - Section 8.3
-Instruction Entry and Editing - Section 8.4
-Output Format and Channel Status - Section 8.5
-Clock and Interface Set-up - Section 8.6
-Output Waveform Display - Section 8.7
-Self Test - Section 8.8
-High Resolution Channel Delay - Section 8.9

Also note that the Configuration Display is available on Power-up and by pressing the [UNIT CONFIG] softkey on the Master Menu.

8.2.2 Explanation of Softkeys - Master Menu

[SELECT UP] - Moves cursor line up to allow menu selection.

[SELECT DOWN] - Moves cursor line in down position to allow menu selection.

[ENT SELECT] - Calls up menu being pointed to by cursor.

[CLEAR ALL] - Resets entire unit, clears all volatile memory and places Master Menu display on CRT. YOU WILL LOSE ALL DATA AND PROGRAMMING LINES!! (Note: when this key is pressed the key will begin to blink. Pressing the key a second time performs the clear function. Pressing any OTHER key disregards the clear command). The [CLEAR ALL] softkey does not clear the internal memory of the 58101 Non-Volatile Memory option. See Section 10.2 58101, Non-Volatile Memory Option.

[C FLAG 0] - Alternate location for setting the C software flag that is used with the conditional branching function of the 58102 Extended Level Looping Option. See Section 10.0, Extended Level Looping Option for information on function and use of this softkey.

[SRQ (ON/OFF)] - Enables or disables the Service Request as specified by the IEEE-488. See Section 11.5, SRQ.

[GOTO LOCAL] - If the PI-5800A is in the Remote Mode, that is, being operated through the IEEE-488 Bus, then this key returns the unit to local operation and re-enables the keyboard. If IEEE Controller has sent Local Lock Out, then this key is not available.

[UNIT CONFIG] - Allows display of the Unit Configuration Display. See Section 8.1, Configuration Display. You must recompile and update the data after pressing this softkey.

8.3 PATTERN ENTRY AND EDITING MENU

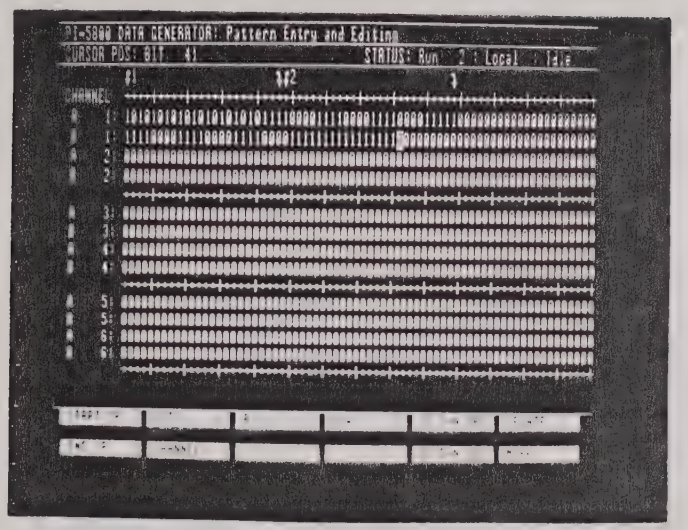


Figure 5. Pattern Entry and Editing Menu

8.3.1 General

The Pattern Entry and Editing Menu provides a display of the individual bit status for the entire system. The display can be scrolled to display all 4,096 bits of memory on each channel up to the maximum of 64 channels. The 1's and 0's are entered with the data keys at the position of the cursor.

This menu is also used for entering data, establishing the size of the subpatterns, setting up mixed format outputs, and many of the PI-5800A's editing features.

The Pattern Entry and Editing Menu is also used for making asynchronous, "on the fly" data changes and for Dynamic Updating, which will provide a synchronized data change during the Run mode. For example: if the PI-5800A Data Generator is in the Run mode, the following procedure will allow you to make an "on the fly" output data change. Place the cursor on the bit that you wish to change and press either [1] or [0] to place the desired data bit into the pattern. Once the desired data changes have been completed, press the [UPDATE] softkey and the data change is made in the output.

8.3.1 General (Cont'd)

The Pattern Entry and Editing Menu is also used for providing bit format status within the pattern. For example, an NRZ output is displayed in normal video, an RZ output is shown in reverse video, and a High Impedance output is shown as dashes in the bit locations instead of the normal 1's and o's. This provides an easy to read status of the output formats that the system is currently providing. The display will not show inverted channels. Refer to the Waveform Display Menu, Section 8.7, for this function.

8.3.2 Subpatterns

A subpattern is a memory field that is defined on the Pattern Entry and Editing Menu, which holds data for use as an output pattern. The subpattern is established between bit positions on the Pattern Entry and Editing display. These bit positions will represent serial strings of data. The patterns can be any length from a minimum of 12 bits to a maximum of the 4,096 bits, or the total channel memory. When the subpattern is established it extends throughout all the channels in the system and is independent of data. The subpattern is like a mail box into which output data will be added later. Figure 6 shows how the subpattern is defined and how it extends through the channels.

Figure 6. Defined Subpattern

Channels	1 Bits 16	
A1	0101010101010101	Channel A1 serial data out
B1	0011001100110011	Channel B1 serial data out
A2	0000111110000000	Channel A2 serial data out
B2	1111000001111111	Channel B2 serial data out
Etc.	Data	Etc. channel serial data

If you do not understand the concept of subpatterns at this point, it is advisable to:

1. View the Pattern Entry and Editing Menu on the PI-5800A.

2. Read Section 8.3.3, Setting Up Subpatterns and the explanation of Pattern Entry and Editing Menu softkeys [START SP] and [END SP], Section 8.3.5.

3. Read or perform, Quick Set-Up, Section 9.0 before continuing.

There can be a maximum of 255 subpatterns established within the 4,096 bits of channel data available. When added as a program statement on the Instruction Entry and Editing Menu, the subpattern can be repeated up to 1,023 times. See [SUBPATTERN] softkey, Section 8.4.2, Instruction Programming, Section 8.4.4.

The following are examples of program lines showing use of the subpattern statement as displayed on the Instruction Entry and Editing Menu:

L3	Subpattern	1 x 10	
L4	Subpattern	3 x 1,023	
Program Line Numbers		Subpattern Numbers	Number of times the subpattern will be repeated on this program line

8.3.3 Setting Up Subpatterns

Here is a sequence of events that demonstrate how to build a subpattern.

1. Press [MASTER] key, then press [CLEAR ALL] softkey twice.
2. Press the [ADVANCE] key to call up the Pattern Entry and Editing Menu.
3. In the Cursor/Menu group of keys, press the [HOME] key two times. Verify that the cursor is located on Bit 1, Channel A1.
4. Press the [START SP] softkey. You should now see a curved arrow and 1 above bit position 1.
5. Using the [CURSOR RIGHT] control, place the cursor at bit position 16.
6. Press the [END SP] softkey. You should now see a curved arrow above bit position 16. (The two arrows define subpattern 1 which now exists between bit position 1 and 16 throughout all the channels in the system.) Refer to Figure 6.

8.3.4 Loading Data into Subpatterns

There are many ways to load data into the established subpatterns.

1. Download from the IEEE-488 bus. (See Programming Section 11.0.)
2. Binary data key entry. (See below.)
3. Algorithmic data entry. (See Algorithmic Data Entry, Section 8.3.7.)

Here is a sequence of events that demonstrates loading data by binary data key entry. IF YOU HAVE NOT SET UP A SUBPATTERN, PLEASE REFER TO SECTION 8.3.3 ON SETTING UP SUBPATTERNS BEFORE CONTINUING.

1. Press the [HOME] key two times. Verify that the cursor is located on bit 1, channel A1.

2. Using the data keys, alternately press data keys [1] and [0] to load data into the subpattern. NOTE: any data entered out of a subpattern structure will not be part of the output pattern.

3. Press the [CURSOR DOWN] key to place the cursor in channel B1.

4. Press the [HOME] key one time. You should now be at bit position one, channel B1. Load the data using data keys [1] and [0].

5. Press [UPDATE] softkey.

6. You are now ready to establish the program to output the pattern you have built. Refer to the Instruction Entry and Editing Menu to set up the program (See Section 8.4).

8.3.5 Explanation of Softkeys

8.3.5.1 Page 1 Softkeys

[START SP] - Defines the start of a subpattern by placing an arrow above the cursor/bit position of the subpattern starting point. Each subpattern established is automatically numbered sequentially at the position established, and extends through all output channels in the system. Minimum subpattern length is 12 bits. The maximum subpattern length is 4,096 bits, or one subpattern which occupies all the data available in the channel.

[END SP] - Defines the end of a subpattern by placing an arrow above a cursor/bit position. This defines the end of the subpattern at this position for all output channels in the system.

[GOTO] - Allows instant access to locations in pattern memory at any bit, channel or subpattern location. Statement examples are as follows:

[GOTO] [START SP] [1] [ENT] - Places cursor at first bit or start of selected subpattern, in this case subpattern 1.

[GOTO] [END SP] [2] [ENT] - Places cursor at last bit or end of selected subpattern, in this case the end of subpattern 2.

[GOTO] [START SP] [2] [+] [2] [ENT] - Places cursor at location of third bit in subpattern 2. Note that the statement, [GOTO] [START SP] [2] [ENT] would place the cursor on the first bit in subpattern 2, so the statement [GOTO] [START SP] [2] [+] [2] [ENT] would place the cursor 2 bits beyond the start of subpattern 2, or at bit position 3 in subpattern 2.

[GOTO] [END SP] [3] [-] [2] [ENT] - Places the cursor at location of third bit from end of subpattern 3. Note that the statement, [GOTO] [END SP] [3] [ENT] would place the cursor on the last bit in subpattern 3, so the statement [GOTO] [END SP] [3] [-] [2] [ENT] would place the cursor 3 bits before the end of subpattern 3.

[GOTO] [CHANNEL] [A] [6] [ENT] - Will place the cursor in the selected channel in the Pattern Entry and Editing Menu display, (the cursor will be in the selected channel at the same bit position it occupied before the command).

[GOTO] [X] [X] [X] [ENT] - Places the cursor at the selected bit location within the 4,096 bit memory. Cursor will be on the same channel as before entry of the statement, but at the new bit location.

[CHANNEL] - Used in conjunction with [GOTO] softkey to instantaneously call up any desired output channel and place it in the Pattern Entry and Editing Menu display. Refer to [GOTO] softkey for example of use.

[RZ] - Softkey used for setting the RZ mode in mixed format outputs. The softkey cycles through [RZ], [RZ (ON)], and [RZ (OFF)]. To convert the entire channel to RZ, see the Output Format and Channel Status Menu, Section 8.5.

[RZ (ON)] - Sets output format to RZ in mixed format modes at bit position of cursor.

(Note: Pressing the [RZ] softkey will cycle the softkey to display [RZ (ON)]. The warning: DATA IN B CHANNEL WILL BE LOST will be displayed. In multiple format mode, the B channel is lost and RZ data can be entered by pressing the [1] and [0] data keys or the [CURSOR LEFT] and [CURSOR RIGHT] keys.
See Section 8.5.3 for description of output format modes.)

In the [RZ (ON)] mode, data can be changed to RZ format by moving the cursor over existing NRZ data which will convert the data to RZ format, or by entering new data at the point of the cursor with the 1 and 0 data keys.

[RZ (OFF)] - Allows removal of RZ data from display, when cursor or the 1 and 0 data keys are advanced through the previously defined RZ data (this area is marked by reverse video). When all the RZ data is removed, the channel will return to its normal state and the B channel will, once again, be available for use. To take the channel out of mixed format mode without removing all the RZ data in that channel, go to the Output Format and Channel Status Menu and place the cursor on the channel you want to change and press the [RZ/NRZ OFF] softkey. This will place the channel in the non-mixed NRZ mode.
See Section 8.5.4 for description of output format modes.

[HI-Z] - Places any "A" channel in High Impedance mode. The "A" channel to be placed in the High Impedance mode is determined by the location of the cursor. Note: Pressing the [HI-Z] softkey will display the following warning with reference to the B channel paired with the "A" channel which will enter High Impedance mode: DATA IN B CHANNEL WILL BE LOST. Pressing the [HI-Z] softkey will now add Hi-Z data. (HI-Z is removed from the output of the channel pair by removing all the dash High Impedance video indicators from the A channel and replacing them with 1's and 0's.) This automatically places the output format in the channel pair to RZ or NRZ depending upon the program in progress. To take the channel out of the High Impedance mode without removing all the dashes present in that channel, go to the Output Format and Channel Status Menu and place the cursor on the channel you want to change and press the [HI-Z OFF] softkey. This will return the channel to its normal state and allow access to the B channel. See Section 8.5.3 for description of output format modes.

[DF DYN UPD] - Defines a block of eight channels to be updated. The number of channel blocks can be any number from one block of eight to the entire system.

[EX DYN UPD] - Executes the synchronous update and pattern change of the channel blocks defined by the [DF DYN UPD] softkeys.

[UPDATE] - Enters any data change displayed on the CRT into the output pattern when pressed. This is an asynchronous "on the fly" change. For example: If an output pattern is established, changing the data as displayed on the CRT will not create a change in the output data until the [UPDATE] softkey is pressed, at which point the new data will replace the existing data at the changed bit positions in the output pattern.

[MORE...] - Calls up and displays second page of softkeys used with the Pattern Entry and Editing Menu. Pressing [MORE...] again will call up and display first page of softkeys.

8.3.5.2 Page 2 Softkeys

[START SP] - Same as [START SP] on Page 1 of softkeys. Included on Page 2 of softkeys for editing convenience.

[END SP] - Same as [END SP] on Page 1 of softkeys. Included on Page 2 of softkeys for editing convenience.

[GOTO] - Same as [GOTO] on Page 1 of softkeys. Included on Page 2 of softkeys for editing convenience.

[DELETE SP] - The statement [DELETE SP] [X] [ENT] will remove the subpattern X from the Pattern Entry and Editing Menu. Note: The pattern must be completed before deleting. If you have begun a pattern pressing the [START SP] softkey, it must be completed by extending the pattern at least 12 bits and pressing the [END SP] softkey before it can be deleted using the [DELETE SP] softkey.

[A+B (ON/OFF)] - The [A+B (ON/OFF)] softkey cascades the A and B channels throughout the system. See Cascade Mode, Section 8.3.8.

[UPDATE] - Same as [UPDATE] on Page 1 of softkeys on Pattern Entry and Editing Menu. Included on Page 2 softkeys for editing convenience.

8.3.6 Editing

This menu also provides a second page of pattern editing softkeys for manipulation of data patterns during set-up. Press the [MORE...] softkey to get second page editing softkeys. The functions available are, delete subpattern, cascade "A" and "B:, and "GOTO" statements. The "GOTO" statements allow call-up of any subpattern location, or any bit location within the channels. Refer to Section 8.3.5.1 for explanation of this soft-key function.

8.3.7 Algorithmic Data Entry

Algorithmic Data Entry allows the entry of repetitious data patterns in a selected channel by an easier method. For example: the following pattern, 1010101010101...1010, where the number of bits is 50 can be entered with the data entry keys in the following manner: [(] [1] [0] [)] [X] [2] [5] [ENT]. You will note that the statement is assembled in the lower left corner of the CRT, directly above the softkeys. The data stream which is 50 bits in length will be inserted in the desired channel at the bit position of the cursor. This method can be used for creating patterns of any length as long as the total number of bits does not exceed 4,096 in any channel.

8.3.8 Cascade Mode

The Cascade Mode allows you to set a single subpattern which will provide up to 8,192 bits of serial data from the "A" channel outputs. In this mode, the "A" and "B" channels are "tied" together and the "A" channel will "toggle" between its 4,096 bits maximum and the 4,096 bits maximum that are established in its paired "B" channel. The Cascade Mode affects all the channels in the system.

To accomplish the Cascade Mode and to allow a synchronous transfer of data, you must have only one subpattern established which can be up to the full channel memory depth. When the Cascade Mode is established, it is in effect for all A-B channel pairs in the entire system, so you are effectively reducing your number of channels available by one-half, but increasing your "A" channel memory by a factor of 2. Only one subpattern pattern can be established in this mode. Pressing the [A+B (ON/OFF)] softkey to the ON position places the unit immediately into the Cascade Mode.

Note: This is an asynchronous change in the output data, so you should be in the stop mode.

8.3.9 Dynamic Update (synchronous)

Dynamic Update replaces the entire contents of an "A" channel with the contents of its paired "B" channel, synchronously. This process can be established for any group of 8 channels, and from a minimum of 8 channels to a maximum of the entire system. The [DF DYN UPD] softkey is used to establish the channel groupings that will be updated by this process. The [EX DYN UPD] softkey performs the actual update. NOTE; DYNAMIC UPDATE OVERWRITES ALL B CHANNEL DATA AT TIME OF UPDATE, AND IT WILL BE LOST.

Note: To execute Dynamic Update there must be at least 8 consecutive bits in the new Channel A Data Pattern that are common with 8 consecutive bits in the old Channel A Data Pattern. These consecutive 8 bits can be located at any locations within the 4,096 bits of data in the channel, assuming you have all 4,096 bits allocated to sub-patterns. The 8 consecutive bits must be located at the same bit positions. For example, if the 8 data bits at bit positions 24 through 31 are the same in the old Channel A data and the new Channel A data, then a legal update can be performed. Every Channel A that is receiving a data change in the update process must have this common section of data AT THE SAME BIT POSITION. If you are changing the data in Channels A1 and A3 and you have selected bit position 24 as the starting point for the 8 consecutive bits, then at bit position 24 in both Channels A1 and A3, the data must be common with the old channel data.

The new data starting bit position 24 in Channel A1 does not have to be the same data pattern as the new data pattern that starts at bit position 24 in Channel A3, but it must be the same as the old Data Pattern in the respective channels.

	BIT POSITION 24
Channel A1 New Data	10111001
Channel A1 Old Data	10111001
Channel A3 New Data	11110000
Channel A3 Old Data	11110000

The above illustrates a correct set of common patterns for the Dynamic Update. Obviously, if you are not making a data change on one of the A channels in the 16-channel block, then all of the data in that channel would remain common so the Dynamic Update would take place correctly.

The actual update will occur on the 5th bit of the first common area, and all the channels on the PC card will be updated at the same time. If you did not change the data in any A Channel on the card, then you will still be running the data on the channel that was previously entered after the Dynamic Update is performed.

The data change that takes place will occur the first time through the first subpattern that contains a data change.

8.3.10 Update (asynchronous)

Since the PI-5800A has a separate set of RAM for the CRT display, it is possible to make pattern changes in the data displayed on the Pattern Entry and Editing Menu without affecting the output. These pattern changes can be added to the output asynchronously by pressing the [UPDATE] softkey. For example: Place the cursor at a bit position you want to change. Change the bit with the [1] and [0] data keys. Press the [UPDATE] softkey.

Note: If any data is changed on the Pattern Entry and Editing Menu, it must be updated in order to be added to the pattern output.

8.4 INSTRUCTION ENTRY AND EDITING MENU

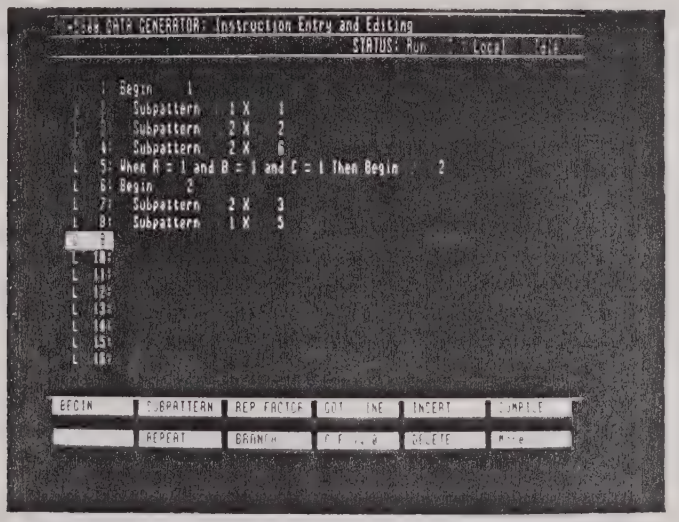


Figure 7. Instruction Entry and Editing Menu

8.4.1 General

The Instruction Entry and Editing Menu is used for establishing the sequence of subpatterns that will be required for a specific pattern application or program.

The menu allows operation of the Data Generator in a very simple form; insert one pattern and run continuously, or in more complex form, where as many as 256 different subpatterns can be intermixed as required. Two levels of looping let you create highly complex patterns in a simple manner that requires only 2 statements. This menu is also used for storage of different pattern group programs that can be selected and run as required, if your unit is equipped with the 58101 Non-Volatile Memory option.

8.4.2 Explanation of Softkeys

[BEGIN] - Assigns a number in sequence to allow differentiation between programs. Begin can be a number from 1 to 256. The first use of the begin statement will be Begin 1, and each successive number in series is assigned automatically. Deleting a Begin statement number removes it from use until the Data Generator is reset by pressing the [CLEAR ALL] softkey on the Master Menu or turning off power. It is advisable to always call up any programs that are stored in EEPROM memory if you have this option, since upon reset, pressing the [BEGIN] softkey will start over at Begin 1. You can use the Begin 1 program that you have created. However, if you enter it into EEPROM memory, it will replace the Begin 1 program in memory if that number has been previously used. Therefore, it is better to always call up any programs in EEPROM memory to verify the Begin numbers that are used, in the event you decide to save the program you are building. [BEGIN] starts a program and [COMPILE] completes it.

[SUBPATTERN] - Is used as part of the program listing to allow you to select the subpattern you wish to run, and to identify the intermixed subpattern sequence. Pressing the [SUBPATTERN] softkey sets a subpattern in the program sequence at the line location defined by the cursor. The subpattern number and repetition rate are then established on the same line. The maximum subpattern number is 255 (the maximum number of subpatterns that can be established) and the maximum repetition factor on a given instruction line is 1,023.

[REPEAT] - Allows any group of subpatterns within a user specified line sequence to be repeated in a single statement up to 255 times. Example: Repeat from line 2 to line 3 X 7. The statement means: Repeat line 2, then 3, 7 times.

[REP FACTOR] - Pressing the [REP FACTOR] softkey sets the number of times a subpattern or repeat statement is to be repeated on a program line. This softkey is pressed to change repetition factors in established programs at the line position of the cursor.

[BRANCH] - If your Data Generator contains the 58102 Extended Level Looping Option, this instruction will allow the establishment of the branch conditions. Refer to Section 10.1 for information on function and use of this softkey.

[C FLAG X] - Alternate location for setting the C software flag that is used with the conditional branching function of the 58102 Extended Level Looping Option. See Section 10.1, Extended Level Looping Option for information on function and use of this soft-key.

[GOTO LINE] - Places the cursor on any selected program line from 1 - 999. For example: [GOTO LINE] [X] [X] [X] [ENT] will place the cursor on line X X X.

[INSERT] - Moves all programming lines down one line from the cursor line position. Used for insertion of additional subpatterns, repeat statements, or begin statements if you are creating an additional program.

[DELETE] - Removes the line at the cursor line position and moves all other program lines up one line.

[COMPILE] - Translates the program that the user has written into code that the PI-5800A can understand. [BEGIN] - starts a program and [COMPILE] completes it. The [COMPILE] softkey is also used to change output programs. If you are running program Begin 2, the following statement will change the output: [COMPILE] [1] [ENT]. This will change the output to the Begin 1 program. When you press the [COMPILE] softkey, a prompt is displayed on the CRT that states: "Enter Begin No. [ENT]." When you enter the begin number and press the [ENT] key, the unit begins to compile the program. If the unit was in run mode at this time, both the pattern output and the clock will stop, then the compile will be performed. Once compile is completed, the unit will begin to run again. If the unit was in the stop mode at the time you pressed compile, it will still remain in stop mode after the compile is completed. The unit will then run the program specified by the begin number entered, when the [R/S] key is pressed. If you enter a begin number for a program that has not been defined (unit has incremented begin 1 and begin 2 and you enter 3), the CRT will display the error, "Invalid BGN NO". Compiling the program in the Run Mode is an asynchronous change in the data output.

8.4.3 Page 2 Softkeys

[MORE] - Calls up Page 2 of softkeys associated with this menu. Pressing [MORE] again will return you to Page 1 of the softkeys.

[STORE] - When the pattern has been established and compiled and is ready to be saved, pressing the [STORE] softkey will begin the sequence of events to allow the save of the required pattern and instruction, if the unit is equipped with the 58101 Non-Volatile Memory Option.

[CLEAR] - Allows individual programs or the entire memory within the memory option to be cleared. Pressing the [CLEAR] softkey will begin the sequence of events to clear programs as necessary. WARNING - YOU WILL LOSE PERMANENT DATA STORED IN NON-VOLATILE MEMORY IF YOU USE THIS KEY!

[RELOAD] - Allows the loading and operation of programs that are internally stored within the 58101 Non-Volatile Memory Option of the PI-5800A. Pressing the [RELOAD] softkey will begin the sequence of events that will allow loading and use of internally stored programs.

[DIR] - Pressing the [DIR] softkey provides a display listing of all programs that are currently stored in the internal memory of the 58101 Non-Volatile Memory Option. Pressing the [DIR] softkey will call up the directory of stored programs. The [DIR] softkey will now be blinking and all other keys on the unit will be locked out. To get back to normal operation, press the [DIR]` softkey again.

[MORE] - The [MORE] softkey allows you to toggle between the Page 1 softkeys for normal operation and the Page 2 softkeys which are provided for use with the 58101 Non-Volatile Memory Option.

8.4.4 Instruction Programming

There are only 4 statements required for programming the PI-5800A.

To start a program, the [BEGIN] softkey is pressed. This action will display the word Begin and the next available number in sequence. If you have begun programming from a [CLEAR ALL] (Master Menu system reset), the first assigned number would be Begin 1. The next step will be to enter the subpatterns that you would like to run in the sequence required. Subpatterns can be intermixed in sequence and repeated as required. Note: To run the program shown below, 2 subpatterns should have been built on the Pattern Entry and Editing Menu.

Here is a sample program to show how it is done:

ACTION TAKEN	PROGRAM LINE/ NUMBER	PROGRAM RESULT
Press [BEGIN]	L1	Begin 1
Press [SUBPATTERN]	L2	Subpattern 1 x 1
Press [1] and [ENT]		
Press [1] and [ENT]		
(You have now completed program lines L1 and L2.)		
Press [SUBPATTERN]	L3	Subpattern 2 x 5
Press [2] and [ENT]		
Press [5] and [ENT]		
(You have now completed program line L3.)		

[RELOAD] - Allows the loading and operation of programs that are internally stored within the 58101 Non-Volatile Memory Option of the PI-5800A. Pressing the [RELOAD] softkey will begin the sequence of events that will allow loading and use of internally stored programs.

[DIR] - Pressing the [DIR] softkey provides a display listing of all programs that are currently stored in the internal memory of the 58101 Non-Volatile Memory Option. Pressing the [DIR] softkey will call up the directory of stored programs. The [DIR] softkey will now be blinking and all other keys on the unit will be locked out. To get back to normal operation, press the [DIR]` softkey again.

[MORE] - The [MORE] softkey allows you to toggle between the Page 1 softkeys for normal operation and the Page 2 softkeys which are provided for use with the 58101 Non-Volatile Memory Option.

8.4.4 Instruction Programming

There are only 4 statements required for programming the PI-5800A.

To start a program, the [BEGIN] softkey is pressed. This action will display the word Begin and the next available number in sequence. If you have begun programming from a [CLEAR ALL] (Master Menu system reset), the first assigned number would be Begin 1. The next step will be to enter the subpatterns that you would like to run in the sequence required. Subpatterns can be intermixed in sequence and repeated as required. Note: To run the program shown below, 2 subpatterns should have been built on the Pattern Entry and Editing Menu.

Here is a sample program to show how it is done:

ACTION TAKEN	PROGRAM LINE/ NUMBER	PROGRAM RESULT
Press [BEGIN]	L1	Begin 1
Press [SUBPATTERN]	L2	Subpattern 1 x 1
Press [1] and [ENT]		
Press [1] and [ENT]		
(You have now completed program lines L1 and L2.)		
Press [SUBPATTERN]	L3	Subpattern 2 x 5
Press [2] and [ENT]		
Press [5] and [ENT]		
(You have now completed program line L3.)		

Press [REPEAT] L4 Repeat line 2 to line 3 X 4
Press [2] and [ENT]
Press [4] and [ENT]

(You have now completed program line L4.)

Press [COMPILE] The flashing indicator will
Press [1] and [ENT] tell you to enter begin
number.
Program is complete.
The basic program shown above would be (SP 1 X 1 + SP 2 X 5) X 4.

The Data Generator will now run subpattern 1, one time, then run subpattern 2, five times, repeating that sequence four times. The instrument will then start over again, after you press the [R/S] key.

Note:

You can intermix subpatterns and use them as required on this menu. However, each repeat statement must stand alone and cannot share any program lines with any other repeat statement, since this would represent a third level of looping. If your Data Generator is equipped with the 58102 Third Level Looping Option, refer to Section 10.1. If you do not have an Output Pattern at this time, refer to Section 9.0, Quick Set-Up and Use.

8.5 OUTPUT FORMAT AND CHANNEL STATUS MENU

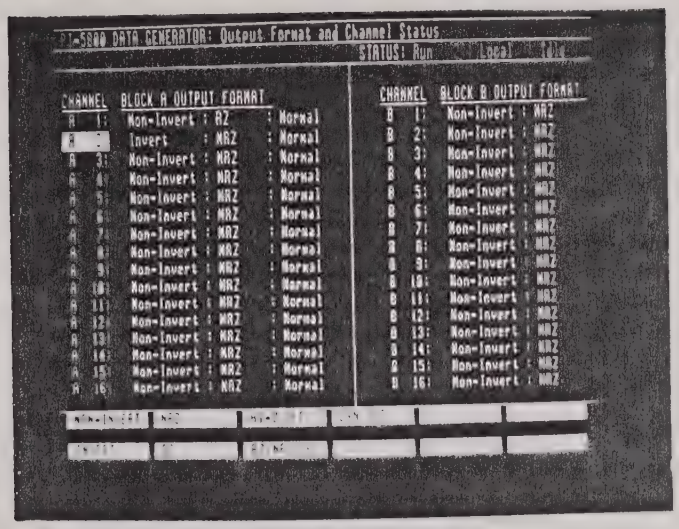

BT-5800 DATA GENERATOR: Output Format and Channel Status
STATUS: Run
CHANNEL BLOCK A OUTPUT FORMAT
A 1: Non-Invert : RZ : Normal
A 2: Invert : NRZ : Normal
A 3: Non-Invert : NRZ : Normal
A 4: Non-Invert : NRZ : Normal
A 5: Non-Invert : NRZ : Normal
A 6: Non-Invert : NRZ : Normal
A 7: Non-Invert : NRZ : Normal
A 8: Non-Invert : NRZ : Normal
A 9: Non-Invert : NRZ : Normal
A 10: Non-Invert : NRZ : Normal
A 11: Non-Invert : NRZ : Normal
A 12: Non-Invert : NRZ : Normal
A 13: Non-Invert : NRZ : Normal
A 14: Non-Invert : NRZ : Normal
A 15: Non-Invert : NRZ : Normal
A 16: Non-Invert : NRZ : Normal
CHANNEL BLOCK B OUTPUT FORMAT
B 1: Non-Invert : NRZ
B 2: Non-Invert : NRZ
B 3: Non-Invert : NRZ
B 4: Non-Invert : NRZ
B 5: Non-Invert : NRZ
B 6: Non-Invert : NRZ
B 7: Non-Invert : NRZ
B 8: Non-Invert : NRZ
B 9: Non-Invert : NRZ
B 10: Non-Invert : NRZ
B 11: Non-Invert : NRZ
B 12: Non-Invert : NRZ
B 13: Non-Invert : NRZ
B 14: Non-Invert : NRZ
B 15: Non-Invert : NRZ
B 16: Non-Invert : NRZ

Figure 8. Output Format and Channel Status Menu

8.5.1 General

This menu displays the output format and channel status. Each channel can be set to be either a complement of the existing data output, non-complement, RZ (return to zero), or NRZ (not return to zero). ALL FUNCTIONS ON THIS MENU ARE BLOCK CHANGES THAT WILL AFFECT THE ENTIRE CHANNEL (See mixed output formatting, Section 8.5.4).

8.5.2 Explanation of Softkeys

[NON-INVERT] - If the output of a given channel is inverted, pressing will change the output to not inverted or normal. The softkey acts upon the channel at the cursor position.

[INVERT] - Inverts the output of the channel at the cursor position.

[NRZ] - Changes the ENTIRE output of the channel at the cursor position in RZ format to NRZ format.

[RZ] - Changes the ENTIRE channel output of the channel at the cursor position in NRZ format to RZ format.

[HI-Z OFF] - Returns the channel at the cursor position to normal.

[RZ/NRZ OFF] - Returns the channel at the cursor position to normal NRZ mode.

[DYN OFF] - Removes all channels from the Dynamic Update condition and returns the Data Generator to normal operating condition. See Dynamic Update section.

8.5.3 Output Formats

RZ - Output follows data, returning to zero during any data 1 output.

NRZ - Output stays high on data 1's until clock leading edge of 0 data output.

RZ/NRZ - This is a mixed-mode format where RZ and NRZ data is present at the same time in the same channel.

HI-Z - High Impedance Output, "floats" between low and high.

TRUE - Normal data output. [Non-Invert]

COMPLEMENT - Inverse of normal data output. [Invert]

With the PI-5800A, you can run (RZ) Return to Zero, (NRZ) Not Return to Zero, RZ/NRZ (a mixed mode of RZ and NRZ on the same output), (HI-Z) High Impedance, true and complement output formats. When running an entire channel in RZ, True, Complement, or NRZ format, the "A" and "B" channels are identical in function and any of the format outputs listed in this paragraph can be run on any channel as long as you are running the format throughout the entire channel.

It is also possible to mix formats on the "A" channels. The PI-5800A is designed to give you this capability by using the "B" channels as the control channels to allow these special applications, hence the channels are set up in "A" and "B" fashion with each complimentary pair sharing the same integer A1 - B1, A2 - B2...

8.5.4 Mixed-Mode Channel Formatting

The PI-5800A allows you to mix channel formats. When mixing 2 formats on an "A" channel, i.e... RZ - NRZ, or RZ - HI-Z, the B channel is sacrificed and becomes a control channel. This mode of operation is established by using the Pattern Entry and Editing Menu. You can use any two of the three formats at a time on any "A" channel, but not all three. However, you could run RZ-NRZ on channel "A1" and RZ - HI-Z on channel "A2" or any combination of 2 on any "A" channel desired. In the mixed format mode, you also have the invert (complement) and non-invert (true) formats available on all "A" channels. Mixed formatting is established with the use of the Pattern Entry and Editing Menu, Section 8.3.

8.5.5 Pulse Width in the RZ Mode

When the PI-5800A Data Generator is using the internal clock in the RZ mode, the RZ pulse width is approximately 12 nanoseconds regardless of the clock's operating frequency, since the pulse width of the internal clock is set at approximately 20 nanoseconds.

In the external clock mode, however, the pulse width of the clock is user defined and the width of the RZ output pulse will be approximately 60% of the duty cycle of the external clock that is being used to drive the system. The external clock must have a minimum pulse width of greater than 10 nanoseconds.

8.6 CLOCK AND INTERFACE SET-UP MENU

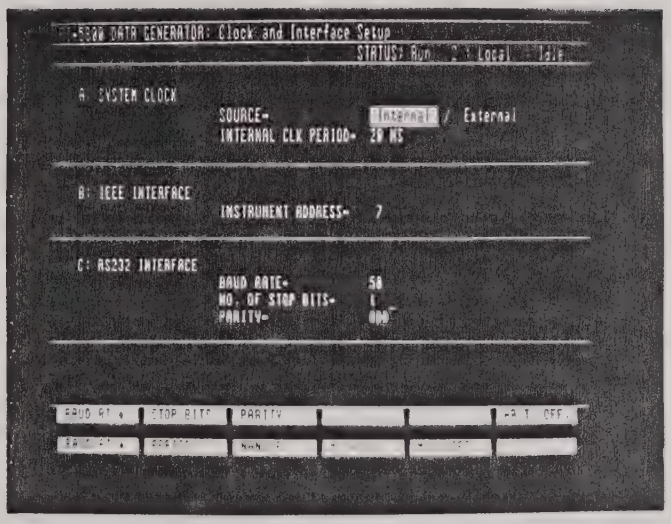


Figure 9. Clock and Interface Set-Up Menu

8.6.1 General

This menu allows selection of clock parameters, the IEEE-488 GPIB instrument address, and the RS-232C information. The clock portion of the menu displays selected internal or external clock, and the internal clock period, in 20 nanosecond increments, from 20 nanoseconds to 1.3107 milliseconds, which will provide clock frequencies from 762.95 Hz to 50 MHz. The external clock input allows the use of any TTL clock in the range from DC to 50 MHz with a 10 nanosecond minimum pulse duration. The PI-5800A clock is programmed to operate at 50 MHz upon start-up, or upon system reset, by pressing [CLEAR ALL] on the Master Menu.

The IEEE-488 GPIB instrument address is set by the coded setting of the DIP switch on the rear panel of the Data Generator. This setting can be from 0 to 31. The selected address number is then displayed on the IEEE-488 interface section of this menu.

The RS-232C section of the menu provides information on the current setting of the Unit Baud Rate. The Baud Rate is adjustable from 50 to 19,200 Baud. You can also select the number of stop bits, and parity.

8.6.2 Explanation of Softkeys

[BAUD RATE UP] - Sequences in ascending order through all available baud rates from 50 to 19,200.

[BAUD RATE DOWN] - Sequences in descending order through all available baud rates from 19,200 to 50.

[STOP BITS] - Allows selection of either 1 or 2 Stop Bits.

[MILLISEC] - For period settings in milliseconds.

[MICROSEC] - For period settings in microseconds.

[NANOSEC] - For period settings in nanoseconds.

[HALT (ON/OFF)] - The Halt mode allows the PI-5800A to be stopped externally by a positive or negative going pulse. The input for the halt is level triggered and requires a TTL input with a minimum pulse duration of 6 nanoseconds. With the unit in the Run Mode and the Halt ON, the PI-5800A will stop if an appropriate signal is fed into the External Halt input on the rear panel (See Section 4.6). Note: When the halt is received, the unit will stop running and the Halt Mode will be turned OFF. You must press the [HALT ON/OFF] softkey to [HALT/ON] if you intend to use the Halt function again.

8.6.3 Setting the Internal Clock Period

The internal clock period is set using the Clock and Interface Set-Up Menu, by using the data keys to set the decimal number and the softkeys to establish the time. The PI-5800A will only establish the internal clock periods in 20 nanosecond increments, so any decimal entry will be rounded off to the nearest 20 nanosecond increment. (If the number 5 is rounded off by the PI-5800A, the result will be higher.) For example: An entry of [PERIOD] [3] [5] [NANOSEC] [ENT] would round off to 40 nanoseconds, which would then be displayed as the internal clock period. The clock period set-up would be as follows:

[PERIOD] [8] [2] NANOSEC] [ENT]

Pressing the [ENT] key will automatically enter the new internal clock period and round it off to the nearest 20 nanosecond increment. The display will now read 80 nanosecond internal clock period. The internal clock period can be set from 1.3107 milliseconds to 20 nanoseconds in 20 nanosecond increments, so your clock frequency steps would be 50 MHz, 25 MHz, 12.5 MHz, etc. If an attempt is made to set a clock period that is less than 20 nanoseconds, the clock period will automatically default to 20 nanoseconds. If an attempt is made to set a clock

period of more than 1.3107 milliseconds, the CRT will display the error message "Period Too Long." All clock period entries must be made in whole number or whole number/decimal form, i.e. [PERIOD] [1] [MILLISEC] [ENT], [PERIOD] [0] [.] [2] [5] [MILLISEC] [ENT]. Entering [PERIOD] [.] [2] [5] [MILLISEC] [ENT] will display a syntax error. Pressing the [CLEAR ENTRY] key will clear any error messages or any errors in data entry. See Section 11.3, Programming Commands.

8.6.4 External Clock Input

The PI-5800A can be driven by an external TTL clock input in the range from DC to 50 MHz. The external clock input is not limited to 20 nanosecond increments, so the clock input can be any desired frequency as long as the minimum pulse width is 10 nanoseconds. If you use the external clock input, make certain that the rear panel switch is in the EXT CLK position.

8.7 Waveform Display Menu

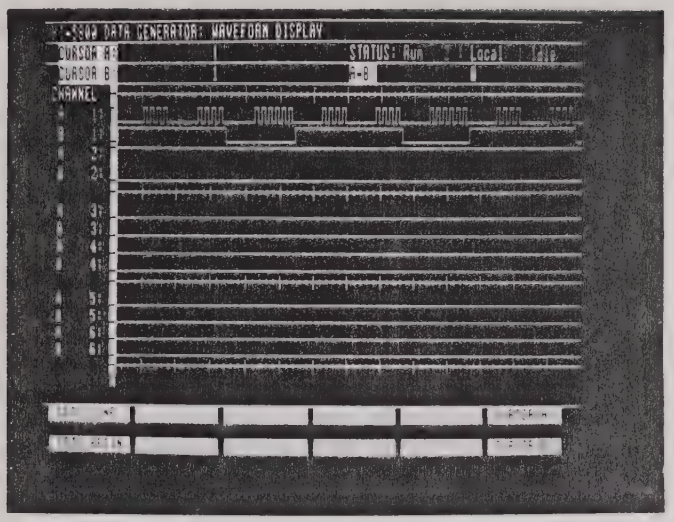


Figure 10. Output Waveform Display Menu

8.7.1 General

This menu displays the output waveforms of the patterns generated on the Pattern Entry and Editing Menu, and a specific program generated on the Instruction Entry and Editing Menu.

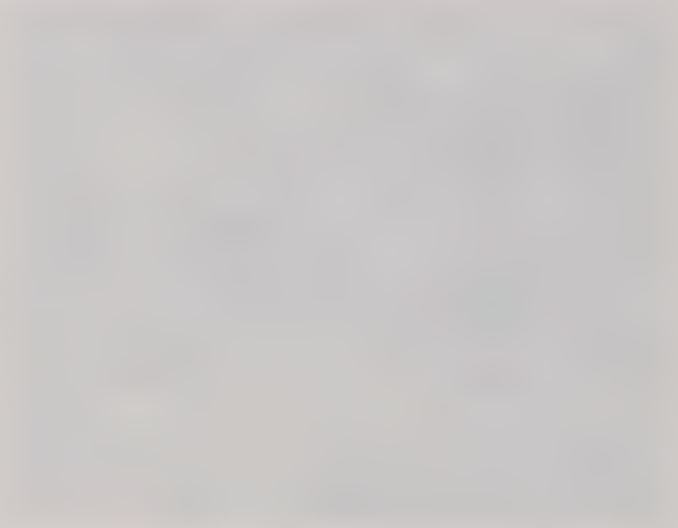
8.7.2 Explanation of Softkeys

[GOTO LINE] - Places the cursor on any selected program line from 1-999 which is generated at "The Instruction Entry and Editing Menu."

[GOTO BEGIN] - Display the output waveforms of the current compiled program beginning at the start of the program.

[CURSOR A] - Blinking when selected.

[CURSOR B] - Blinking when selected.



8.8 SELF TEST MENU

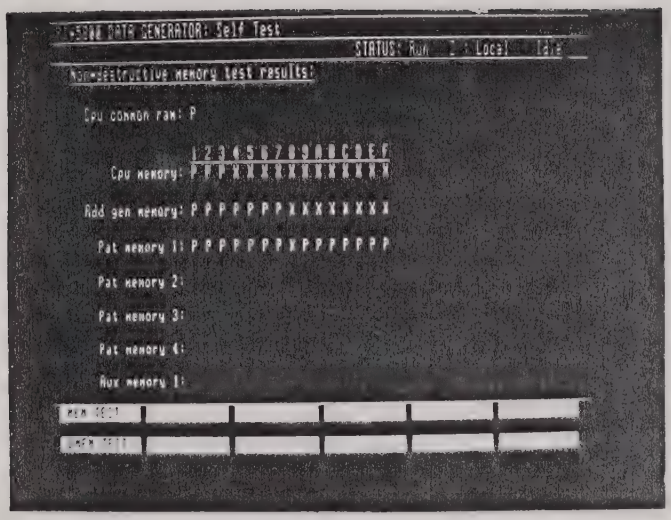

Non-destructive memory test results:
Cpu common ram: P
Cpu memory:
Add gen memory: P P P P P P P X X X X X X X
Pat memory 1: P P P P P P P X P P P P P P P
Pat memory 2:
Pat memory 3:
Pat memory 4:
Aux memory 1:

Figure 11. Self Test Menu

8.8.1 General Description

The self test feature provides a quick functional test of a large portion of the circuitry inside the PI-5800A. Two softkeys allow selection of tests that verify operating integrity at various critical locations. The bank numbers refer to specific memory locations and the display indicates either a Pass (P), Fail (F), or Don't Care (X) condition exists.

8.8.2 Explanation of Softkeys

[MEM TEST] - Tests most of the unit's memory locations. This is a non-destructive test (no loss of RAM information).

[DMEM TEST] - Conducts a more thorough test of memory which includes all address points. This is a destructive test (loss of information in RAM). This test takes approximately 10 minutes to complete.

8.9 HIGH RESOLUTION CHANNEL DELAY MENU

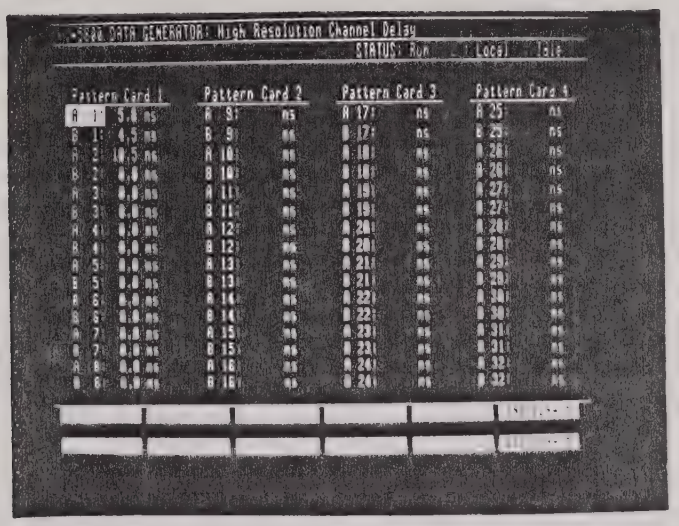
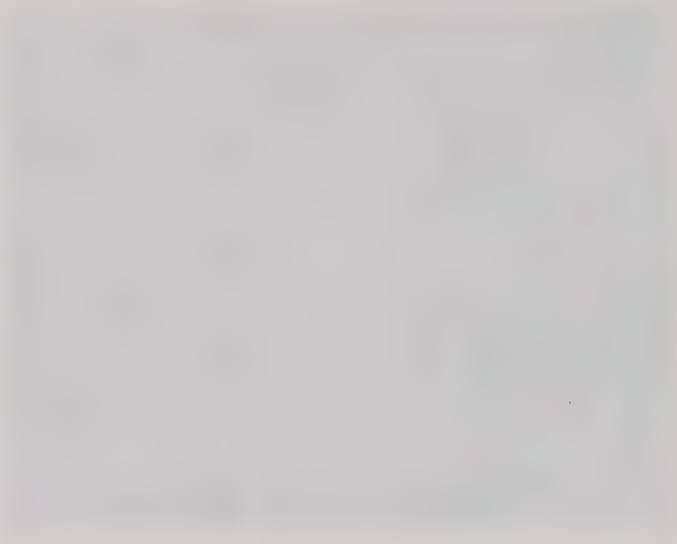


Figure 12. High Resolution Channel Delay Menu

8.9.1 General Description

The High Resolution Channel Delay Menu provides a status board to facilitate adjustment of the output skew between channels in units that are equipped with the Pulse Instruments 58002 16-channel output cards with programmable delay. The output delay in each channel is adjustable over a range of 31.5 nanoseconds. The adjustments allow incrementing ([INC DLY .5]) or decrementing ([DEC DLY .5]) the delay in 500 picosecond increments on the channel at the cursor position.

Since the total delay available on each channel that is equipped with the programmable delay option is 31.5 nanoseconds, this is effectively an infinite delay since 31.5 nanoseconds will place the pulse back into the next clock. The delay can be adjusted during the Run Mode. Since the specifications for the delay setting are 500 picoseconds ±300 picoseconds, all delay settings are empirical values, and should be measured for accuracy and correctness of desired setting.



You will note that the [INC DLY .5] and [DEC DLY .5] have a range of 31.5 on the display of the Programmable Delay Menu. This number represents the actual delay, with 0 being no delay and 31.5 being the maximum.

The Delay that is set skews the entire 4,096 bits of the channel with respect to all other channels in the system.

8.9.2 SETTING THE PROGRAMMABLE DELAY OVER THE IEEE-488 BUS

The Programmable Delay is not dumped along with the instructions and data when a file is saved. However, it can be set over the IEEE-488 Bus when the file is up loaded, or "on the fly".

When using the softkeys on the Programmable Delay Menu to set the delay, the maximum increment is 31.5. Over the IEEE-488 Bus these setting numbers are doubled, so the range of settings is from 0 to 63., with 0 being a setting of no delay.

The following command illustrates the setting of the Programmable Delay over the Bus:

"DELAY A1-36"

In the above statement, A1 refers to the channel number and 36 represents the doubled setting of the original delay.

8.9.3 Explanation of Softkeys

[INC DLY .5] - Pressing this softkey adds 500 psec delay from the channel at the cursor position.

[DEC DLY .5] - Pressing this softkey removes 500 psec delay to the channel at the cursor position.

9.0 QUICK SET-UP AND USE

This section is designed to give you the basics of operation of the PI-5800A in a condensed manner, by setting up the unit and running a pattern.

Equipment Required

1 BNC cable approximately 4 feet in length

150 MHz Oscilloscope

16-Channel output cable (P/N 88000050) for connection from the Output Connector J-1 to oscilloscope.

PI-5800A Data Generator

1. Plug the 16-channel output cable into connector J-1. Refer to Section 4.7 for J-1 pin out configuration. (If you do not have the P/N 88000050 output cable, you will require 2 BNC terminated 50 ohm coaxial cables which will be connected to channels "A1" and "B1".)

2. Connect channels "A1" and "B1" to the oscilloscope.

3. Verify that the Clock Select switch on rear panel is in the INT CLK position, and that the Cascade/Normal switch is in the Normal position.

4. Connect BNC cable from TRIGGER OUT on rear panel to EXT trigger input on oscilloscope.

5. Verify that front panel power switch is in the OFF position.

6. Plug AC power cord into rear panel power receptacle, and connect to power source.

7. Turn on unit power. After a few seconds, you will see the Unit Configuration display. This menu provides information on what PC boards and features are contained within your particular unit. (Press any key and the Master Menu will be displayed.)

8. Press the [ADVANCE] key to call up the Pattern Entry and Editing Menu.

9. In the Cursor/Menu group of keys, press the [HOME] key 2 times. Verify that the cursor is located on Bit 1, Channel A1.

10. Press the [START SP] softkey. You should see a curved arrow and 1 above bit position 1.

11. Using the [CURSOR RIGHT] key, place the cursor at bit position 16.

12. Press the [END SP] softkey. You should see a curved arrow above bit position 16.

13. Press the [HOME] key once to place the cursor back at bit position 1.

14. Using the Data Entry keys, alternately press data keys [1] and [0] to load data in Channel A1 between the subpattern arrow indicators out to bit position 16.

15. Press the [CURSOR DOWN] key to place the cursor in channel B1.

16. Press the [HOME] key one time. You should now be able to load data in channel B1 between bit position 1 and bit position 16.

17. Press the [ADVANCE] key one time to display the Instruction Entry and Editing Menu.

18. Verify that the cursor is at position L1. If not, press the [HOME] key or move the cursor to line L1.

19. Press the [BEGIN] softkey. The display now reads Begin 1 on line L1.

20. Press the [SUBPATTERN] softkey. Subpattern is now displayed on line L2.

21. Press the [1] data key and [ENT]. The subpattern is now defined as subpattern 1.

22. Press the [1] data key and [ENT]. You have now set the repetition factor at 1.

23. Press the [COMPILE] softkey.

24. Press the [1] data key and [ENT]. You have now instructed the Data Generator to translate the Begin 1 program.

25. Press [MASTER] key to call up the Master Menu.

26. Press the [ADVANCE] key one time to call up the Pattern Entry and Editing Menu.

27. Press the [UPDATE] softkey. This function will set the data into the pattern generator.

28. Press the [RUN/STOP] key. The Data Generator should now be in Run mode and the pattern entered in Step 14 should be displayed on the oscilloscope, when triggered properly.

10.0 OPTIONS FOR THE PI-5800A

10.1 58102 EXTENDED LEVEL LOOPING OPTION

10.1.1 General

The 58102 Extended Level Looping Option is comprised of two major features of benefit to users of the PI-5800A Data Generator. The first feature is Third Level Looping and the second feature is Conditional Branching. Please note that these features cannot be used at the same time in the same program. Also note that if you are using the branching option in a program that is built under one Begin number, you cannot branch into another program that is using the third level looping function. These two features must be used separately. To understand this section of the operator's manual, you must be familiar with the sections of the manual describing the Pattern Entry and Editing Menu and the Instruction Entry and Editing Menu, and have a basic knowledge of programming the PI-5800A Data Generator. Even if your main interest is in using the branching function and not the Third Level Looping function, we urge you to read the Third Level Looping section first, since much of the information you will need to know regarding the branching function is common to the Third Level Looping option.

10.1.2 THIRD LEVEL LOOPING

Third level looping allows the data output from the data generator to be looped an additional level. This function is set up in the same manner as the second level repeat statement. The following program demonstrates use of the Third Level Looping:

```
L1  Begin 1

L2  Subpattern 1 x 1

L3  Subpattern 2 x 3

L4  Subpattern 3 x 2

L5  Repeat from L3 to L4 x 2

L6  Subpattern 4 x 1

L7  Repeat from L3 to L6 x 3
```

In the program above, "Begin 1" starts the program and establishes a reference number to allow selection of this particular program when more than one program is established. L2, L3 and L4 establish a sequence of three subpatterns that will be output from the Data Generator.

L5 is a second level repeat. Once the Data Generator runs the subpatterns established on lines L2, L3 and L4, the repeat statement on line 5 will re-run lines 3 and 4 the required number of times, in this case 2 times.

Line 7 is the Third Level Looping statement and repeats the entire program portion from lines 3 to 6. There are certain restrictions on the program lines that can be included in the Third Level Looping statement. THIRD LEVEL LOOPING CAN BE ASSIGNED TO REPEAT FROM PROGRAM LINES THAT ARE THE BEGINNING LINES OF SECOND LEVEL LOOP STATEMENTS AND FROM THE FIRST LINES AFTER BEGIN STATEMENTS. You must include all the lines that are within any of the second level loop statements that will be part of the Third Level Loop statement and you can include as many second level statements as you require. For example, if your program included four second level loop statements, you could select the first line of any of these second level loop statements to begin the third level loop. The Third Level Loop must include all of the lines covered in the second level loop statement. You cannot begin the third level with only a portion of the lines in any second level loop included.

For example, in the program below:

```
L1  Begin 1
L2  Subpattern 1 x 2
L3  Subpattern 2 x 3
L4  Subpattern 3 x 4
L5  Subpattern 5 x 1
L6  Repeat from L4 to L5 x 2
L7  Subpattern 4 x 3
L8  Subpattern 2 x 2
L9  Subpattern 5 x 2
L10 Subpattern 6 x 7
L11 Repeat from L8 to L10 x 2
```

The above program contains Second Level Repeat Statements on lines 6 and 11. We could establish a Third Level Loop statement on Line 12 that could begin at lines 2, 4 and 8, since by our previous definition these lines represent either the first lines of second level repeat statements or the first lines after Begin statements. An example of one possible Third Level statement that could be included in this program for line 12 would be:

```
L12  Repeat from L4 to L11 x 2
```

This line represents a Third Level Loop that would include two Second Level Statements.

However, there are other lines that can be included in the Third Level Loop. These are lines that are grouped together and not part of Second Level statements and can be as few as one line if the one line is surrounded by lines that are part of the second level repeat statements.

There is an imaginary second level loop of 1 that consists of lines 2 and 3, and another imaginary Second Level Loop of 1 that consists of line 7. These imaginary loops exist by the fact that if the program executes these lines and no repeat statement is connected with them, then in fact they represent a repeat of one, and, since our definition of the beginning point of any Third Level Loop is any line that is the first line of a Second Level Loop, we can include the first lines of these imaginary Second Level Statements into our definition. In fact, the internal programming of the PI-5800A handles these lines in this manner.

To verify this point, you could place a second level repeat statement on line 8 as follows:

L8 Repeat from L7 to L7 x 1

This line could be added by using the [INSERT] softkey. You would also have to change the references in line 11 to the new line numbers to maintain the existing loops as previously established.

Note that the program will operate in the same manner as before, since the repeat of one was already implied. So, you can see that by using a repeat of 1, you can perform a Third Level Loop from any line that would be followed by a "repeat of one" statement since it would satisfy our definition of a line that could be used in a third level loop.

Therefore, it is possible to establish Third Level Loops that begin at lines 2 and 7, as well as the previously established lines, 2, 4 and 8.

Although line 2 had previously qualified since it was the first program line following a Begin statement, it also qualifies since it is the first line of an imaginary Second Level Loop.

You can repeat the selected segment of the program up to 128 times in a Third Level Looping statement. You can repeat a Second Level Looping segment up to 256 times. A subpattern can be repeated on any line 1,024 times.

The following simple program provides an example of the three different levels of looping using the maximum number of loops for each level:

```
L1  Begin 1

L2  Subpattern 1 x 1,024

L3  Repeat from L2 to L3 x 256

L4  Repeat from L3 to L4 x 128
```

In the program above, subpattern 1 will be repeated a total of 33, 554, 432 times. L2 is an example of First Level Looping, L3 is an example of Second Level Looping, and L3 is an example of Third Level Looping.

10.1.3 CONDITIONAL BRANCHING

The conditional branching option allows the establishment of conditional branches between programs built under different Begin statements. For example, a branch established in a program labelled Begin 1 can branch to a program built under Begin 2 if certain conditions are met. These conditions involve the use of flags to set the conditions and transfer program control from one program to another.

There are eight possible conditions that can be established with the Branching Option. These conditions involve the use of two hardware flags and one software flag. The flags are labelled A, B, and C. The A and B flags are hardware set by the application of TTL level voltages, and the C flag is set by software. The hardware flags are set on pin number 19 for the A flag and pin number 24 for the B Flag, of the RS-232-C connector, and the software flag is available on the Instruction Entry and Editing Menu as the [C Flag (0)] or [C Flag (1)] softkey. The default upon power-up for the A and B flags is high. See Section 11.9 for information on programming this option. For pinouts, see Section 4.6.

If you elect to use a flag in other than don't care state, you must completely define the conditions of all other possibilities within the program. For example:

```
L1  Begin 1
L2  When A=X and B=0 and C=1 Then Begin 1
L3  When A=X and B=1 and C=1 Then Begin 2
L4  When A=X and B=1 and C=0 Then Begin 1
L5  When A=X and B=0 and C=0 Then Begin 2
```

In the above example, all conditions of the B and C Flags are listed so that the Data Generator understands how to handle all possibilities in order to allow successful branching. Had A Flag been any condition except don't care, the possibilities for A would have also needed explanation within the program. Remember to place all branching conditions in the Begin 1 program and Compile the Begin 1 program before running.

The [C FLAG (0)] or [C FLAG (1)] softkey is also available on the Master Menu. You can also assign flag A, B or C, don't care status (X) with the [-] data key. When programming don't care over the IEEE-488 bus, the asterisk (*) is used for don't care. See Sectioni 11.9.

The flags are set to the on or off condition by the application of a TTL high or low level to the proper pins on the RS-232-C connector.

Setting the voltage level to high or approximately 2.4 vdc represents the high or 1 level. Setting the voltage low or approximately Ovdc represents the 0 or low level. The Default setting of the A and B flags upon power-up is High. The Default on power-up of the C flag is 0.

The levels of the software flag are set by toggling the [C FLAG (X)] softkey to either [C FLAG (1)] for high level or [C FLAG(0)] for low level.

Since there are three flags available, there are eight possible combinations which can be established.

The eight conditions are shown below:

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

You can also use the Don't Care function (X) for flags A, B or C. X is set with the [-] data key, when entered from the front panel and with the asterisk (*) when used over the IEEE-488 bus or RS-232 port.

For example, in the first condition shown above the A, B and C flags are all set to zero. This is defined by the voltage on pin 19 (the A flag) being set to 0. The voltage on pin 24 (the B flag) being set to 0, and the C flag is set to 0 on the Instruction Entry and Editing Menu by toggling the [C FLAG (X)] softkey.
The instructions for the conditional branching are entered on the Instruction Entry and Editing Menu by pressing the [BRANCH] softkey. The following program demonstrates use of the Branch function:

```
L1  Begin 1
L2  When A=X and B=X and C=0, then Begin 2
L3  When A=X and B=X and C=1, then Begin 3
L4  Begin 2
L5  Subpattern 1 x 1
L6  Begin 3
L7  Subpattern 2 x 1
```

In the above program, the branching functions are placed in the Begin 1 program. The branches are established by pressing the [BRANCH] softkey on the Instruction Entry and Editing Menu. The cursor will then be to the left of A, and by pressing the [1] or [0] or [-] data key followed by [ENT], the value of the A flag will be assigned and the cursor will move to the B flag position. Set the value of the B and C flags in the same manner as A, and when the entire program is entered, press [COMPILE] [1] [ENT] to compile the program and [UPDATE] on the Pattern Entry and Editing Menu to update the data. Now press the [R/S] key to run the program. Any line can be edited by deleting the line and re-entering it, with the [DELETE] and [INSERT] softkeys.

When the program is in Run mode, it will branch to Begin 2 and begin running subpattern 1. To run the program built under Begin 3, press the [C FLAG (0)] softkey to display [C FLAG (1)] and note that the program halts, recompiles and runs the program built under Begin 3. The Halt, Recompile and Program change requires less than one-half second.

If, in the above program, the conditions had not been met for the branch to Begin 3, the program would have stayed in the program established under Begin 2. Since the C flag was set at 0, and the A and B flags were set Don't Care (X), the Begin 2 program would continue to run until the conditions exist as written on line 3, (assuming you were using the Run/Stop mode instead of the Single Cycle mode).

The A and B flags must be held at the 1 or 0 level throughout the time that you require the branch. For example, if applying a high TTL level to the A flag branches the program, removing the high level will return the unit to the program that was running before the branch, since removing the high TTL level would reset to the previous branching condition.

The Branch is controlled by hardware, and it is immediate and the program stops in one clock cycle. The Data Generator then recompiles and runs the branch program selected.

When compiling programs containing branch statements, you should compile only the first program containing a branch statement.

The commands and programming information for this option over the IEEE-488 Bus are in Section 11.0, Programming the PI-5800A.

10.2 58101 NON-VOLATILE MEMORY OPTION

10.2.1 General

The 58101 Non-Volatile Memory Option allows the user to store up to 48K bytes of instrument set-up and instructions within the PI-5800A Data Generator for later recall. The following instructions will assume that you are familiar with operation and set-up of the PI-5800A as described in Sections 8.3, 8.4 and 9.0 of this manual.

The option does not allow set-up of clock frequency or delay settings if the 58002 Delay Option is being used.

Although the messages displayed on the Instruction Entry and Editing Menu refer to EEPROM memory, the actual devices being used are battery RAMS, which were selected for their faster load and save access times. The battery RAMS contain lithium batteries within the IC's and have an expected life use of 10 years.

The memory option will allow storage of up to eight begin statements (programs), or up to 48K bytes of information, whichever occurs first. This option typically will allow 2 large programs or up to 8 of lesser size, depending on the number of patterns, pattern channels, and instructions to be stored. More than one program can be called up at a time with reservations. For more information on this point, see Section 10.2.8, Non-Volatile Memory Conservation.

None of the functions of this option are accessable over the IEEE-488 Bus or RS-232C compatible lines. Any program that is currently stored internally by the memory option can be accessed over the bus once it is loaded and readied for use.

All of the softkeys required for this option are located on Page 2 of the Instruction Entry and Editing Menu. You may select this menu by pressing the [MORE] softkey located on Page 1 of this menu. The softkeys that are located on Page 2 of the Instruction Entry and Editing Menu are as follows:

10.2.2 Explanation of Softkeys - 58102 Non-Volatile Memory Option

[STORE] - When the pattern has been established, updated, compiled and is ready to be saved, pressing the [STORE] softkey will begin the sequence of events to allow the save of the required pattern and instruction.

[CLEAR] - Allows individual programs or the entire memory within the memory option to be cleared. Pressing the [CLEAR] softkey will begin the sequence of events to clear programs as necessary. WARNING - USE OF THIS SOFTKEY WILL RESULT IN PERMANENT LOSS OF DATA STORED IN NON-VOLATILE MEMORY.

[RELOAD] - Allows the loading and operation of programs that are internally stored within the PI-5800A. Pressing the [RELOAD] softkey will begin the sequence of events that will allow loading and use of internally stored programs.

[DIR] - Pressing the [DIR] softkey provides a display listing of all programs that are currently stored in the internal memory. Pressing the [DIR] softkey will call up the directory of stored programs. The [DIR] softkey will now be blinking and all other keys will be locked out. To get back to normal operation, press the [DIR] softkey again.

[MORE] - The [MORE] softkey allows you to toggle between the Page 1 softkeys for normal operation and the Page 2 softkeys which are provided for use within the 58101 Non-Volatile Memory Option.

10.2.3 Clearing the Entire 58102 Memory Option - Internal Memory

WARNING - USE OF THIS SOFTKEY WILL RESULT IN PERMANENT LOSS OF DATA STORED IN NON-VOLATILE MEMORY.

When you receive the PI-5800A or begin saving programs for the first time, you should first clear the Non-Volatile Memory as follows:

1. To clear the entire Non-Volatile Memory, go to the Instruction Entry and Editing Menu, and press the [MORE] softkey to display the Page 2 Non-Volatile Memory group of softkeys.

2. Press the [CLEAR] softkey. The message will display "Clear: Enter Begin No. <ENT>".

3. Press 99 on the data keys followed by [ENT]. The message will now display "EEPROM STORAGE SPACE REMAIN 48K BYTES". 48K bytes is the condition of maximum memory available.

10.2.4 STORE FUNCTION

Once you have built a program that you want to save, you should proceed as follows:

1. On the Instruction Entry and Editing Menu, press the [MORE] softkey to display the Non-Volatile Memory group of softkeys.

2. Press the [DIR] softkey to verify that there is not a program existing in memory with the same begin number that you have used to build the program that you want to save. If this condition exists, refer to Section 10.2.7, Changing Begin Numbers. You should also refer at this time to Section 10.2.8, Non-Volatile Memory Conservation.

The following steps begin the save function:

1. Press the [STORE] softkey. The message will display "STORE: ENTER BEGIN NO. <ENT>."

2. Enter the begin number of the program that you want to store. Press the [ENT] data key.

3. The message will now display "STORE: HOW MANY CHANNELS <ENT>".

4. Enter the number of channels you want to store, and press [ENT]. Note: You should save only the channels that you want to use in your program. For better conservation of memory, do not save unused channels.

5. The message will now display "EEPROM STORAGE SPACE XX BYTES." This is your indicator that the program has been saved and also, the amount of uncommitted memory remaining.

To verify the save, press the [DIR] softkey. The screen will now display the programs that are in memory, including the one that you have just saved.

Please note that the [DIR] softkey is blinking and all other keys on the unit are locked out to your access. To return to normal operation, press the [DIR] softkey once more. This will take you back to normal mode and will display the programs and the instructions that are currently loaded and ready for use.

10.2.5 LOADING PROGRAMS FROM THE NON-VOLATILE MEMORY OPTION

1. On the Instruction Entry and Editing Menu, press the [MORE] softkey to call up Page 2, the Non-Volatile Memory group of softkeys.

2. Press the [DIR] softkey to display the directory of programs currently stored in Non-Volatile memory. Select the begin number of the program that you wish to enter.

3. Press the blinking [DIR] softkey to remove all keys from lock-out.

4. Press the [RELOAD] softkey. The message will display "RE-LOAD: ENTER BEGIN NO. <ENT>".

5. Enter the begin number of the program that you wish to load, and press [ENT].

The instructions for the program you have requested will now appear on the screen. To view the actual patterns at this time, press the Master Menu key followed by pressing the Advance key. You should now be on the Pattern and Entry Editing Menu.

6. Press the [UPDATE] softkey on the Pattern Entry and Editing Menu to load the data.

7. Press the [ADVANCE] key to get to the Instruction Entry and Editing Menu.

8. Pressing [COMPILE], followed by the begin number of the loaded program, followed by enter, will place the Data Generator in operation.

9. To run the program, press the [R/S] key in the operational group of keys.

10.2.6 To Clear Individual Programs From the Non-Volatile Memory

WARNING - USE OF THIS SOFTKEY WILL RESULT IN PERMANENT LOSS OF DATA STORED IN NON-VOLATILE MEMORY.

It is possible with the Non-Volatile Memory Option to clear individual programs as well as the entire Non-Volatile Memory.

To clear an individual program, go to the Instruction Entry and Editing Menu. Press the [MORE] softkey to display the Page 2 Non-Volatile Memory group of softkeys, and perform the following:

1. Press the [CLEAR] softkey; the message will display "Clear: Enter Begin No. <ENT>".

2. Press the Begin number of the program that you want to delete or clear followed by [ENT].

3. The message will now display "EEPROM STORAGE SPACE REMAIN XXK BYTES". The program has now been deleted.

4. Press the [DIR] softkey to verify that the program has been erased. Re-pressing the blinking [DIR] softkey will return you to normal operation, and pressing the [MORE] softkey will return you to the Instruction Entry and Editing Menu, Page 1 softkeys.

10.2.7 To Change Begin Numbers

If you enter a begin number that does not correspond to an existing program, the message will read "Invalid Begin Number" - press any key to continue.

If you save a program under a begin number that has already been used in the internal memory option, you will erase the previously stored program and replace it with the program that you have just entered.

If you find that you have built a program under a begin number that already exists in the internal memory, and you want to save both the existing program and the program that you have just built, you can do so by changing the begin number of the newly created program as follows:

1. On the Instruction Entry and Editing Menu with the program that you want to change the begin number to save, displayed, (not the directory), perform the following steps.

2. Place the cursor on the line containing the begin number that you want to change.

3. Press the [DELETE] softkey.

4. Press the [INSERT] softkey. You should have a blank space where the begin statement was previously listed.

5. Press the [BEGIN] softkey. You should now have the program under a new begin number.

6. If you find by checking the directory that this begin number is also being used, repeat the above process until you obtain an unused begin number under which the program can be saved.

7. Now recompile the program by pressing the [COMPILE] softkey and entering the new begin number.

8. Go to the Pattern Entry and Editing Menu and update the data by pressing the [UPDATE] softkey.

9. You can now return to the Instruction Entry and Editing Menu to begin the save of the program.

10.2.8 Non-Volatile Memory Conservation

In order to obtain maximum benefit from the Non-Volatile Memory Option, the following steps should be taken when saving or loading more than one program at a time.

The Non-Volatile Memory Option saves the program as established under the begin number that was used on the Instruction Entry and Editing Menu to build the program as well as the number of channels of pattern data that you require from the Pattern Entry and Editing Menu.

When you reload programs (up to the maximum of 8), the instructions on the Instruction Entry and Editing Menu are all loaded in sequence under their respective begin numbers, and the menu could display the instructions for all eight programs if all the programs that could be stored were loaded in sequence.

However, since the Pattern Entry and Editing Menu can only display one set of data, and since the generator can only output one set of data, the data that is displayed on the Pattern Entry and Editing Menu is the data that corresponds with the data that was saved with the last program that you have called up.

For example, if you built 3 programs that shared common pattern data, and you saved the 3 programs completely, it would occupy a large amount of memory. Note that the save routine asks you "How many channels do you want to save." If you are using 16 channels for each of the 3 programs that you save, then you have saved 48 channels of data. This represents a tremendous waste of memory, since the 16 channels are common to all three programs. A better method would be to save only one channel with each of the first 2 programs that you save (the unit cannot save 0 channels), then save the 16 channels of pattern with the third program. Then, if the third program that contains all the pattern data is loaded last, you have achieved your result at a cost of only 18 channels saved (the 16 you need plus the one per other program that is sacrificed), versus 48 channels by the previous method.

11.0 PROGRAMMING THE PI-5800A

11.1 GENERAL

The PI-5800A is programmed for use on the IEEE-488 bus through the use of simple statements. These statements are listed in Section 11.3.

Below you will find a sample program showing the steps required to output a pattern that is very similar to the pattern set-up that is described in Section 9.0, Quick Set-Up and Use.

The program listed does not include your controller commands on the string set-up, but lists the actual commands and the output sequence required.

***** NOTE; COMMANDS DESCRIBED IN THIS SECTION CAN BE *****
***** SENT TO THE PI-5800A IN LOWER CASE OR ALL CAPITAL *****
***** LETTERS. ALSO, TERMINATE ALL COMMAND LINES WITH *****
***** A <CARRIAGE RETURN>, <LINE FEED>, AND AN EOI. *****
***** TRANSMISSIONS FROM THE PI-5800A TO THE CON- *****
***** TROLLER WILL BE TERMINATED SIMILARLY. WHEN *****
***** INPUTTING DATA FROM THE PI-5800A, EOF, [I.E. *****
***** END OF FILE (1A HEX), <CARRIAGE RETURN>, <LINE *****
***** FEED>, AND AN EOI. SEE IEEE-488 STANDARD FOR YOUR *****
***** CONTROLLER.] WILL BE SENT, WHEN THE UNIT *****
***** HAS COMPLETED SENDING THE REQUESTED DATA. *****
***** SPACES ARE IGNORED. IT SHOULD ALSO BE NOTED THAT *****
***** THE PI-5800A WILL NOT RESPOND TO ANY PROGRAMMING *****
***** COMMANDS IF THERE IS AN ERROR PENDING. IF THE SRQ *****
***** IS ENABLED, THEN A SERIAL POLL WILL CLEAR OUT ANY *****
***** ERROR STATUS.IF THE SRQ IS DISABLED,THEN A STATUS *****
***** COMMAND WILL CLEAR OUT THE ERROR. *****

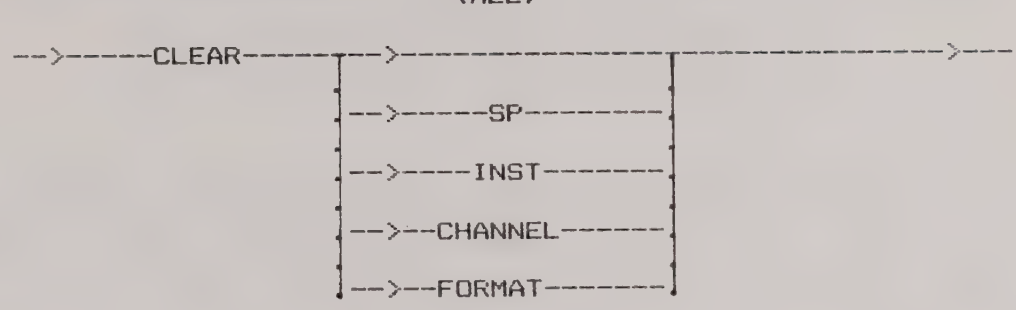
COMMAND LINES	FUNCTION
CLEAR	Clears memory and resets Data Generator
DEFINE SP 1 (1-16)	Sets up subpattern 1 starting at bit position 1 and being 16 bits long
1 BEGIN 1	Includes program line and establishes Ref.# for program
COMPILE 1	Sets finish of program, and tells Data Generator that program is entered and ready to run. See Section 8.4.2.

LOAD CHANNEL A1	Selects channel that will be loaded with data
(10110011) * 1	Channel A1 data entry. See Section 11.2.1
UPDATE	Sends data into pattern memory
RUN	Runs pattern continuously

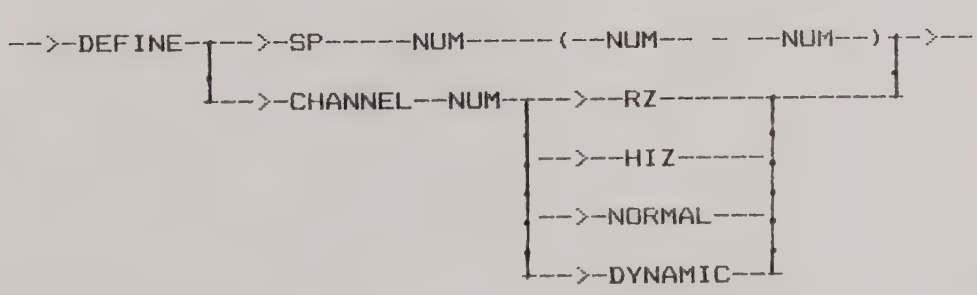
11.2 BINARY DATA ENTRY

In Binary Data Entry, the data is entered at the position of the cursor by enclosing the data to be entered in parenthesis and adding a multiplier. For example, the pattern 11011110 would be entered at the cursor position by sending the following command to the PI-5800A: (11011110) * 1. The pattern could have been entered 100 times with the following statement: (11011110) * 100. For more information on this method of data entry, see the sample program listed in Section 11.6.

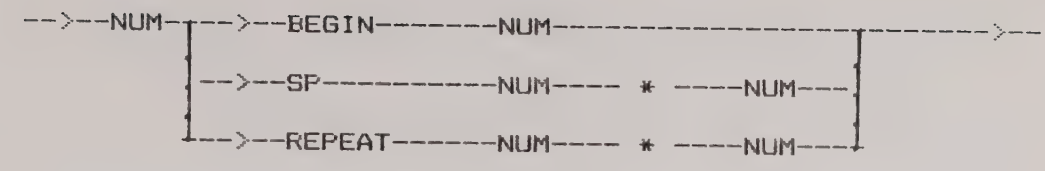
11.3 PROGRAMMING COMMANDS



The CLEAR is the system master reset. It clears out all volatile memory and resets the system when the CLEAR is the only command. When used in conjunction with the lower paths it will clear each item as required. (NOTE: SP = Subpatterns, INST - Instructions.)



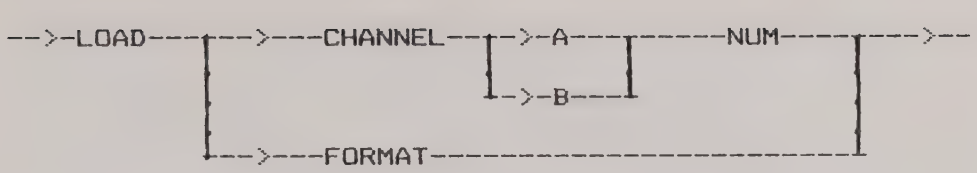
DEFINE sets up the subpattern number and bit locations. Example: DEFINE SP 3 (17 - 33) sets subpattern 3 to start at bit 17 and be 33 bits long. DEFINE is also used to set up the mixed-mode operations, i.e. RZ/NRZ and HI-Z. (Note: RZ = RZ/NRZ in this context.) See also Section 11.4.1.



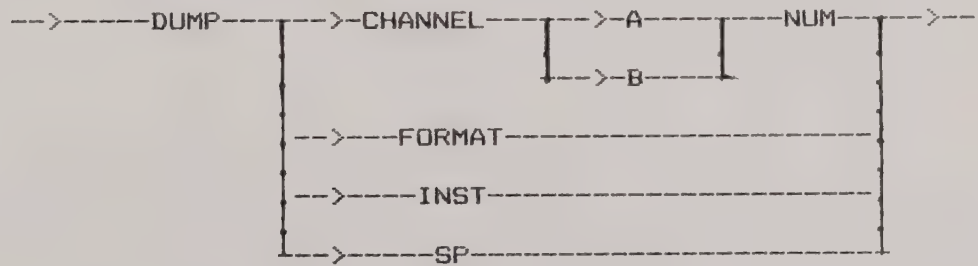
The statements above are used in establishing the actual program as it appears on the Instruction Entry and Editing Menu. The beginning NUM is used for setting the line number. The Begin statement is used for identifying a particular program, and the next two paths are used for setting the subpattern and repeat statements into the program. The first NUM in the SP path defines the subpattern number to be used. The second NUM in the SP path is the repetition factor for that particular subpattern. The first NUM in the REPEAT path is the "FROM" number (See Section 8.4). The second NUM in the repeat path is the repetition factor for that particular instruction. Here is an example of their use:

```
1 BEGIN 1
2 SP 1 * 5
3 SP 2 * 3
4 REPEAT 2 * 3
5 SP 1 * 1
6 SP 3 * 3
7 REPEAT 6 * 1
```

COMPILE translates the series of commands referenced by the begin number (NUM) into instructions that the PI-5800A can understand.

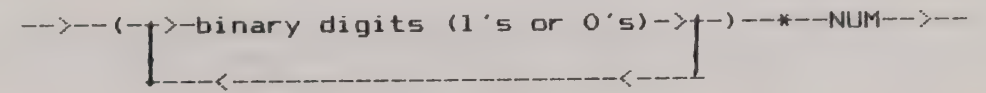


LOAD prepares the selected channel, for receiving data patterns, or loads the block format (not to be confused with Binary Data Loading). LOAD puts the PI-5800A into a state where it is waiting for data to be passed to it in the form of Intel ASCII Hex. In order to better understand the form of the data expected in the use of this command, it is recommended that the user try the DUMP command. The PI-5800A dumps out data in Intel ASCII Hex format.



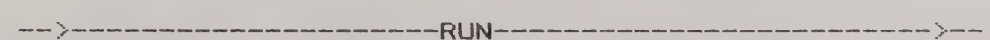
The DUMP command will prepare the PI-5800A to dump out the data associated with the form of the command used. If the command DUMP CHANNEL A1 is sent, the PI-5800A will be prepared to send out the data stored in channel A1 upon request. The data will be in the form of 16 Intel ASCII Hex strings. The end of the transmission will be identified by an EOI (See IEEE-488 Standard.)

Binary Data Entry

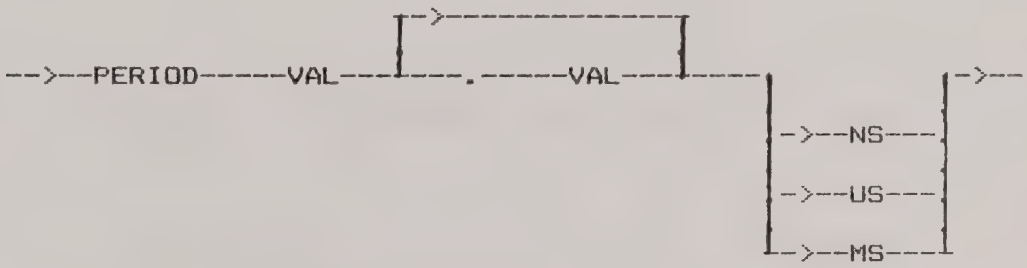


This command enters binary data starting at the present location of the cursor. Use the GOTO command to position the cursor. See Sections 11.2 and 11.4.

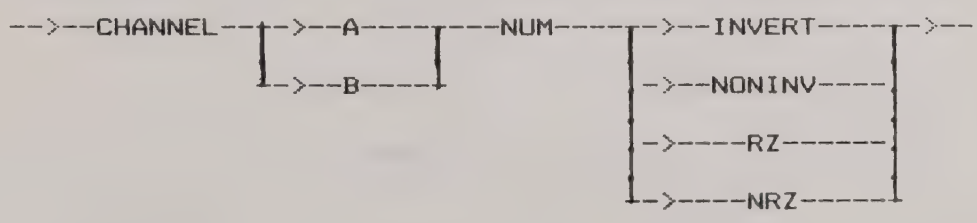
Places the data pattern into pattern memory to be run.



RUN starts the pattern output. The RUN command places the unit in an automatic mode and it will continue to output and repeat the compiled program until the STOP command is sent.

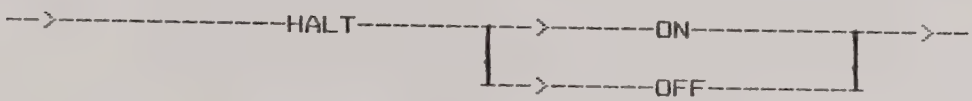


The internal period is set using the above formula. The term VAL represents a group of digits which hold the whole and fractional parts of the value of the period to be set. NS, US, and MS are nanoseconds, microseconds, and milliseconds, respectively. (See Section 8.6.3 for details regarding the range of values that can be set. See Section 11.6, line 560 for an example.)

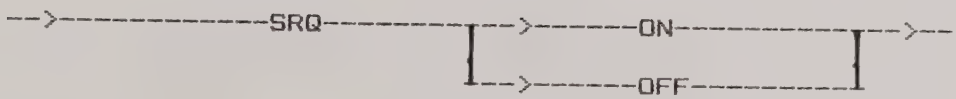


This command sequence is used for setting any selected channel's output format. See Section 8.5.3. (NOTE: NONINV = Non-Invert.)

Used to place the PI-5800A in or out of Cascade Mode. See Section 8.3.8.



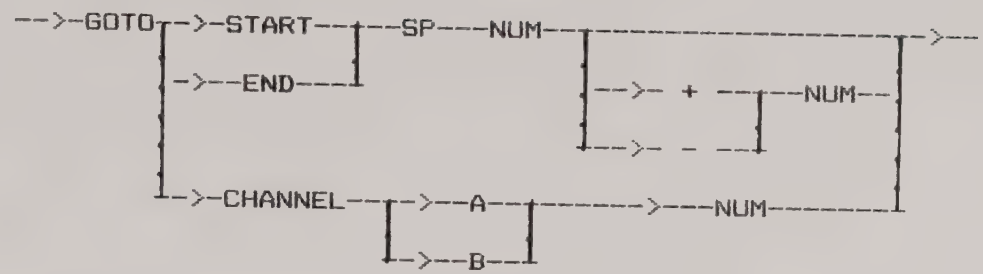
HALT turns on or off the external halt function. (See Section 8.6.2.)



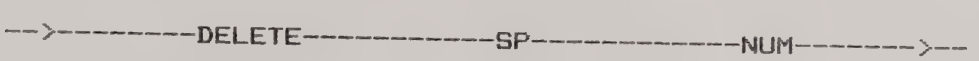
Turns the Service Request ON or OFF. Note that the PI-5800A will respond to a Serial Poll by transmitting the error code which generated the Service Request. (See IEEE-488 Standard and Section 12.0.)

Provides status of error messages or information on the Run/Stop condition. Send this command to the PI-5800A, then "INPUT" the data into your controller. This command clears out any error that might be pending. NOTE: THE PI-5800A CEASES TO RESPOND TO PROGRAMMING COMMANDS WHEN THERE IS AN ERROR PENDING.

11.4 Editing Commands



The GOTO statement allows quick access to various channel, bit, and subpattern locations. See Section 8.3.5.1. This command moves the cursor to the desired location in the Pattern Entry and Editing Menu.



The DELETE SP command will delete any selected subpattern by removing it from the Pattern Entry and Editing Menu.

11.4.1 Programming Mixed Format Modes

The following examples describe the set-up required for programming mixed formats over the IEEE-488 Bus. In the first program, the bit locations from 10 to 16 will be set in RZ mode, while the remainder of channel A1 will be in the NRZ mode. When defining modes across the bus that involve an A and B channel pair, only the channel pair number is used, so Define channel number would be correct instead of Define A1, B1, etc. An example of this difference would be shown by viewing the differences between the commands for Dump and the commands for Define in Section 11.3. This definition for Define is also true for the Dynamic Update.

To set bits 10 to 16 in RZ mode and the balance of the channel in NRZ mode, the commands would be as follows:

"Define channel 1 RZ"
"Goto channel B1"
"Goto 10"
"(111111) * 1"
"Update"

The above commands would set the A1, B1 in the RZ mode and select the portion of the A1 channel that would be driven by the B1 channel.

To set RZ for the channel pair, use 1 as the drive data in the B channel. To set Hi-Z for the channel pair, use 0 as the drive data in the B channel.

Setting the same channel pair Hi-Z and NRZ would be as follows:

```
"Define channel 1 Hi-Z"
"Goto channel B1"
"Goto 10"
"(000000) * 1"
"Update"
```

Notice that "0" is used to drive the Hi-Z mode, since the B channel is set to all 1's, while "1" is used to drive the RZ mode, since the B channel is set to all 0's.

11.5 Service Request (SRQ)

7	6	5	4	3	2	1	0
	SRQ			EXT HALT	SC END	IDLE	ERR

Figure 13. The 8 Bits of the Service Request Byte

The above figure shows the assigned bits that comprise the service request Byte. If any of bits 0 through 3 are set, it will also set the service request (SRQ) bit 6.

The following occurances will set the SRQ bit:

ERR (BIT 0)

If any error occurs during the operation of the PI-5800A, the err bit and the SRQ bit will be set. When the controller recognizes the request for service, a serial Poll must be performed to reset the err bit and the SRQ bit.

Sending the word "STATUS" will return the error number. The error numbers are explained in the operating manual in Section 12.0.

IDLE (BIT 1)

The Idle bit and the SRQ bit are set after the PI-5800A has performed a dynamic update. See the information on dynamic update for instructions on performing this function. A serial Poll will automatically reset the Idle bit and the SRQ bit.

SC END (BIT 2)

When operating the PI-5800A in the program mode, using the single cycle function, the previously compiled program is run one time through. After the completion of this single cycle, the SC END bit and the SRQ bit are set. Performing a serial Poll will reset the SC END bit and the SRQ bit and allow you to continue. See Section 7.2, Operational Controls.

EXT HALT (BIT 3)

When the external Halt is triggered, the EXT HALT bit and the SRQ bit are set and the output from the PI-5800A is stopped. When serial Poll is performed the EXT HALT bit and the SRQ bit are reset and you are ready to continue. To reset the External Halt input to receive the next External Halt pulse, the command "HALT ON" must be sent after the serial Poll.

11.6 SAMPLE PROGRAM FOR PI-5800A (written in BASIC)

```
10   REM     ************************************************
20   REM     *                                              *
30   REM     *  This is a sample program which will         *
40   REM     *  demonstrate how to program the PI-5800A      *
50   REM     *  DATA GENERATOR.                             *
60   REM     *                                              *
70   REM     ************************************************
80   REM
90   REM     In line 110, the IEEE address is 7.
100  REM
110  OPEN 1, 7
120  REM
130  REM     The following three lines define 3 subpatterns: the
140  REM     first starting at bit position 1 and being 20 bits
150  REM     long, the second starting at bit position 20 and
160  REM     being 51 bits long, etc.
170  REM
180  PRINT #1, "DEFINE SP1(1-20)"
190  PRINT #1, "DEFINE SP2(20-51)"
200  PRINT #1, "DEFINE SP3(100-1000)"
210  REM
220  REM     The following four lines define a program to output
230  REM     the subpatterns in the following order: (SP1 X 1) +
240  REM     ((SP2 X 3) + (SP3 X 1)) X 5.
250  REM
260  PRINT #1, "1 BEGIN 1"
270  PRINT #1, "2 SP1 * 1"
280  PRINT #1, "3 SP2 * 3"
290  PRINT #1, "4 SP3 * 1"
300  PRINT #1, "5 REPEAT 3 * 5"
310  REM
320  REM     The Next line will compile the above set of
330  REM     instructions into a language that the machine can
340  REM     understand.
350  REM
360  PRINT #1, "COMPILE 1"
370  REM
380  REM     The next few lines will load some data into
390  REM     channels A1 and B16.  This is meant only as
400  REM     an example.
410  PRINT #1, "GOTO CHANNEL A1"
420  PRINT #1, "GOTO 1"
430  PRINT #1, "(00000000111010101010101010101010100000) * 1"
440  PRINT #1, "GOTO CHANNEL B16"
450  PRINT #1, "GOTO 10"
460  PRINT #1, "(011) * 20"
470  REM
480  REM     The next line will update the pattern memory
490  REM     with the data just entered.
```

```
500  REM
510  PRINT #1, "UPDATE"
520  REM
530  REM      The next line will set the internal period to
540  REM      302.1 microseconds.
550  REM
560  PRINT #1, "PERIOD 302.1 US"
570  REM
580  REM      The next line starts the DATA GENERATOR.
590  REM
600  PRINT #1, "RUN"
610  REM
620  CLOSE 1
630  END
```

11.7 PLACING THE DATA GENERATOR IN THE DYNAMIC UPDATE CONDITION

When Dynamic Update is to be performed, the entire PC card that contains the channel to be updated is placed in the Dynamic State. See Section 8.3.9 for additional information.

WARNING: YOU WILL LOSE ALL DATA IN THE B CHANNELS WHEN THIS UPDATE IS PERFORMED. For example, if you want to update channel A1, you would start by sending the following command over the bus.

"DEFINE CHANNEL 1 DYNAMIC"

This command will place 16 channels, or all channels on the same card as Channel A1, in the dynamic condition. This example would therefore include Channels A1 through A8, and Channels B1 through B8.

In this function, as in using the mixed format modes, the channels work together in pairs A1-B1, A2-B2, A3-B3, etc... and since you are in the dynamic condition, you cannot place the cursor in any B channel on the PC card that is in the Dynamic Mode. All data changes will occur through the A channels. In our example we are Dynamic Updating the A1 channel PC card so the program would continue as follows:

"DEFINE CHANNEL 1 DYNAMIC" - This command sets up the card for the Dynamic Update, and places a copy of the Channel A data in its respective B paired channel on the card. Note that the A in the channel number is not required. See Section 11.4.1.

"GOTO CHANNEL A1" - Sets the cursor on Channel A1.

"GOTO 1" - Sets the cursor on Bit 1.

"(1101101101101101)" - Loads new data in bits 1 through 16 in Channel A1. (Note: this will not affect the patterns being output at this time).

"DYNAMIC UPDATE" - Changes the A1 Channel data to the new patterns.

"DEFINE CHANNEL 1 NORMAL" - Takes Data Generator out of Dynamic Mode.

You can place any PC card in the Dynamic Condition by placing the cursor on the selected channel and defining it Dynamic. You can remove the Dynamic Condition from the PC card by placing the cursor in any A channel and defining it normal.

11.8 SETTING THE PROGRAMMABLE DELAY OVER THE IEEE-488 BUS

The programmable Delay is not dumped along with the Instructions and Data when a file is saved. However, it can be set over the IEEE-488 Bus when the file is up loaded, or "on the fly".

When using the softkeys on the Programmable Delay Menu to set the delay, the maximum increment was 31.5. Over the IEEE-488 Bus these setting numbers are doubled, so the range of settings is from 0 to 63., with 0 being a setting of no delay.

The following commands illustrate the setting of the Programmable Delay over the Bus:

"DELAY A1-36"

In the above example, A1 is the channel number, and 36 is the delay setting.

11.9 PROGRAMMING THE 58102 EXTENDED LEVEL LOOPING OPTION

The following commands will allow use of the Extended Level Looping Option over the IEEE-488 Bus.

Command	Function
NUM----REPEAT----NUM---- * ----NUM	Third Level Loop (Same statement as Second Level Loop) i.e...3 Repeat 2*3

The Third Level Looping function is set over the bus in the same manner as Second Level Looping. The differentiation between the commands is established by the line numbers referenced in the program. The NUM before the Repeat statement represents the program line number.

NUM---BRANCH(NUM-NUM-NUM)---NUM	Conditional Branch i.e...3 BRANCH (*10) 2

The Branch statement includes the A, B and C flags in parenthesis and the Begin number following. In the above statement, A is Don't Care, B = 1, and C = 0. The NUM before the word Branch in the above example represents the program line number. When Don't Care is sent over the bus the asterisk (*) is used in place of the dash (-) that represents Don't Care when front panel keyboard entry is used.

C FLAG----NUM	Setting of software flag C, i.e...C FLAG 1 or C FLAG 0

The software C Flag can be set and toggled over the bus with the command above.

12.0 ERROR AND WARNING MESSAGES

12.1 GENERAL

Below is a list of the PI-5800A's error messages and warnings. When an error occurs or a warning is necessary, the message will be displayed on the menu you are presently using. The error message is removed by pressing any key; the warning message is removed by pressing any key except [ENT].

In the program mode, the errors and warnings are transmitted to the controller in the form of the NUMBERS associated with the error messages specified in Section 12.2, below. Sending STATUS over the bus will return the code corresponding to the messages listed below. ALL ERRORS must be cleared in this manner before proceeding.

12.2 EXPLANATION OF ERROR AND WARNING MESSAGES

1. B CHANNEL WILL NOT TRI-STATE - An attempt was made to place a B channel in High Impedance mode. Since the B Channel is used for control, it cannot be used in High Impedance mode. Place the cursor in the appropriate A Channel. See Sections 8.3.6 and 8.5.4.

2. DATA IN B CHANNEL WILL BE LOST - When setting an A Channel into Mixed Format mode, the paired B Channel acts as a control Channel. Once the warning is issued, if you press [ENT], the B Channel will go into control mode and immediately "overwrite" ony data in B Channel memory. See Section 8.5.4 for a disscussion of mixed formats, and Sectioni 8.3.6 for discussion of [RZ (ON/OFF)] and [HI-Z] softkeys.

3. INVALID FORMAT - An attempt was made to change output format from Hi-Z or RZ/NRZ, to RZ/NRZ or Hi-Z. Any change from Hi-Z or RZ/NRZ formats must be made through Normal mode (See Section 11.3.). Go from Hi-Z or RZ/NRZ mode to Normal, then make the required change. See Sections 8.3.6, 8.5.4, and 11.3.

4. B CHANNEL WILL NOT RETURN TO ZERO - An attempt was made to set the B Channel in Mixed Format mode. Since the B Channel is used for control, it cannot be used in Mixed Format mode. Place the cursor in the appropriate A Channel. See Section 8.5.4.

5. NUMBER TOO LARGE - An attempt was made to enter a number out of range. Normally occurs when trying to go to a subpattern above 255, or trying to enter a programming line greater than 999.

6. SYNTAX ERROR-INPUT IGNORED - Normally occurs when the wrong instruction sequence was entered or upon receipt of a command word which is mis-spelled.

7. SUBPATTERN START NOT DEFINED - An attempt was made to enter a subpattern end without a subpattern start being defined. Establish the start of the subpattern by placing the cursor at the required bit position and pressing the START SP softkey. See Section 8.3.3. Also, if an attempt is made to GOTO a subpattern start that is not defined, this error will result.

8. OFFSET ERROR - An out-of-bounds condition normally displayed when using GOTO SUBPATTERN START or END routines. An attempt was made to go to a bit position that was above or below the 4,096 bit channel mamory.

9. CHANNEL MEMORY FULL - All 4,096 bits in the channel have been committed. There is no more memory available in the channel.

10. OVERFLOW - An attempt was made to enter a number larger than 65,535.

11. NUMBER TOO SMALL - An attempt was made to enter a number less than 0.

12. MINIMUM SUBPATTERN LENGTH IS 12 - An attempt was made to enter a subpattern with less than 12 bits which is the minimum subpattern size allowable. If you require a pattern of less than 12 bits, build it into another pattern. For example, if you require a subpattern of 16 bits to run 10 times followed by a subpattern of 3 bits that will run one time, you can build the 16-bit subpattern to run nine times, then build the same 16-bit pattern and add the 3 bits. Now run your new 19-bit pattern one time and you have accomplished your task. See Section 8.3.2 and 8.3.3.

13. SUBPATTERN END NOT DEFINED - An attempt was made to enter a subpattern with a start position established, but no end to the pattern. Establish the end of the subpattern by placing the cursor at the required bit position and pressing the [END SP] softkey. See Section 8.3.6. Also, if an attempt is made to GOTO a subpattern end that is not defined, this error will result.

14. COMPILE ERROR - An attempt was made to compile a program which is not able to be compiled. See Sectioni 8.4.3.

15. CAN'T GOTO THAT CHANNEL - An attempt was made to GOTO a B Channel that is presently being used as a control channel for its paired A Channel. Go to the appropriate A Channel.

16. NO LISTENERS - The IEEE-488 controller has given the PI-5800A the command to "Talk" and there is no instrument on the bus in the "Listen" mode. See Section 11.0.

17. CHECKSUM ERROR - The Checksum that has been computed by the PI-5800A on the last Intel ASCII Hex transmission that it had received, has not matched up with the checksum that came across with that data transmission.

The following errors (18 through 22) are compile errors associated with the programs generated in the Instruction Entry and Editing Menu (See Section 8.4 for details.):

18. INVALID BEGIN NO. - The program that has been asked to be compiled does not exist. Only compile defined Begin Numbers.

19. NO DEFINED SUBPATTERNS - A compile has been attempted, but no subpatterns had been previously defined. Define the subpatterns that are used in the particular program to be compiled.

20. INVALID FROM NO. - Within a REPEAT instruction in the program to be compiled, the "FROM" number (See Sectioin 8.4.2) is impossible to accomplish within the limits of two-level looping.

21. SHOULD BE SUBPATTERN INSTRUCTION - An inappropriate instruction has been entered within a program.

22. INVALID SUBPATTERN NO. - A compile has been attempted of a program which attempts to use a subpattern which is not yet defined.

23. UPDATE ERROR - Occurs whenever there is an error in data transfer between CPU Ram and Pattern Ram. This error could occur during Update or Dynamic Update.

24. PERIOD TOO LONG - An attempt was made to enter an Internal Clock Period that exceeded 1.3107 milliseconds. Internal Clock Periods must be entered within the range of 20 nanoseconds to 1.3107 milliseconds in 20 nanosecond increments. See Section 8.6.3.

25. NO MATCH FOUND FOR DYNAMIC UPDATE - An attempt was made to execute a Dynamic Update without at least 8 bits in the A and B channel being dynamically updated being common. See Section 8.3.10 Dynamic Update (synchronous).

26. DYNAMIC UPDATE AND CASCADE WILL NOT WORK TOGETHER - You cannot combine Dynamic Update mode and Cascade mode at the same time.

27. REPEAT LEVEL IS MORE THAN ALLOWED - An attempt was made to program a level of repeat that does not exist.

28. PARITY ERROR - Indicates that a character was received with incorrect parity when "with parity" or "force parity" is enabled.

29. FRAME ERROR - Indicates that the stop bits have not been detected.

30. RECEIVER OVERRUN - Indicates that the previous character in the receive holding register has not been read by the CPU and that a new character has been loaded into the receive holding register.

31. TRANSMITTER EMPTY - Indicates that the transmitter has underrun, i.e., both the transmit holding register and transmit shift register are empty.

32. TRANSMIT BREAK - Indicates the PI-5800A receives a command to transmit a break.

33. RECEIVED BREAK - Indicates that an all zero characater of the programmed length has been received without a stop bit.

13.0 RS-232-C COMPATIBILITY

13.1 GENERAL

The PI-5800A Data Generator is compatible with the EIA STD. RS-232-C standard for half duplex asynchronous receiver/transmitter operation when connected to a controller employing serial binary data interchange.

The informationi regarding use of this standard is given below. However, we suggest that you refer to the manual on your controller for implementation instructions and information before proceeding.

13.2 RS-232-C PIN OUTS

The pins on the RS-232-C connector required for this function are as follows:

Pin 1 - Protective Ground (shield)

Pin 2 - Receive Data (xmit data connection on controller)

Pin 3 - Transmit Data (receive data connection on controller)

Pin 5 - Clear to Send

Pin 6 - Data Set Ready

Pin 7 - Signal Ground

13.3 PROGRAMMING INFORMATION

Control of the PI-5800A is accomplilshed through use of the commands XON and XOFF. XON [ASCII(17)], usually in the form CHR$(17) is used by the PI-5800A to tell the remote device to start sending. XOFF [ASCII(19)], usually in the form CHR$(19) is used by the PI-5800A to tell the remote device to stop sending.

The parameter set-up for the RS-232-C function is located on the Clock and Interface Set-up Menu. The soft keys are used to toggle through the available settings until the needed parameters are selected.

The default settings upon power up are:

1200 - Band Rate

NONE - Parity Bit

8 - Data Rate

1 - Stop Bit

These settings can be changed by selecting the appropriate softkey on the clock and interface set-up menu and toggling the selected softkey through the available settings until the required setting is displayed.

Once you have set up the required RS-232-C parameters and have finished all other programming functions that you require prior to operating the PI-5800A over the RS-232-C link, press the [MASTER] key and return the Data Generator to the Master Menu display. The PI-5800A handles all RS-232-C functions only in the Master Menu mode.

All commands that are used on the PI-5800A to operate the instrument over the IEEE-488 Bus are identical to the command requirements through the RS-232-C port.

NOTE: If the baud rate is set for operation at higher than 2400 bps, the device communications buffer in your controller must be set for more than 1K bytes. For example, when using IBM PC/XT Basic, the command:

Basic A/C: 1024

will set the buffer for 1024 bytes.

14.0 EPROM REPLACEMENT IN THE PI-5800A

14.1 GENERAL

The following procedure explains the steps necessary to replace the EPROMs in the PI-5800A. In the event of a software update, or improvement in software function, you will be sent a replacement set of EPROMs. The following procedure will aid you in EPROM replacement. Please note that software improvements will rarely affect your computer program. In the event that a change does affect your program, you will receive detailed information on the modification and where it will affect your program. If you do not receive these instructions, the change in software will not affect your computer programming.

1. Remove 6 screws in top cover and lift off top cover.

2. Remove the 2 screws securing the board retaining bracket.

3. The CPU board that contains the PROMS is the full length board closest to the CRT. Remove the rear board retaining screw and carefully remove the board from the socket.

4. With the board on the bench, component side up, position the board so that the connector pin side is closest to you. The PROMS that you are going to replace are the labelled devices in the upper right hand corner of the board. Make certain that the numbered devices are inserted in the same location as the numbers of the devices that are removed. Verify that all devices are installed in their proper sockets and that all are firmly pressed down.

5. Re-insert the CPU board, turn on power and verify proper operation of the unit.

6. Install the board retaining screw on the rear of the CPU board.

7. Install the front board retaining bracket and verify that all boards are in their proper slots in the bracket. Secure with 2 screws.

8. Install top cover.